
Análisis biomecánico para ciclistas



Trabajo de Fin de Grado de Ingeniería Informática

Carlos Jaynor Márquez Torres

Laura Pérez Jambrina

Daniel Tocino Estrada

Dirigido por

Sonia Estévez Martín

Luis Fernando Llana Díaz

Facultad de Informática
Universidad Complutense de Madrid

Madrid, 16 de Junio de 2017

Análisis biomecánico para ciclistas

Trabajo de Fin de Grado

Carlos Jaynor Márquez Torres

Laura Pérez Jambrina

Daniel Tocino Estrada

Dirigido por

Sonia Estévez Martín

Luis Fernando Llana Díaz

Facultad de Informática

Universidad Complutense de Madrid

Madrid, 16 de Junio de 2017

Autorización

Se autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria como el código, la documentación y/o el prototipo desarrollado.

Carlos Jaynor Márquez Torres

Laura Pérez Jambrina

Daniel Tocino Estrada

Agradecimientos

No queremos finalizar este proyecto sin dar las gracias a todas las personas que nos han ayudado y apoyado a lo largo de su realización y que nos han acompañado a lo largo de nuestra carrera universitaria.

Agradecer primeramente a nuestra familias, todo su apoyo durante estos años, sin ellos hubiera sido totalmente imposible llegar hasta aquí. Gracias por vuestro amor incondicional, por creer en nosotros en todo momento y por brindarnos la posibilidad de aprender y formarnos no solo profesionalmente sino como personas. Es el mejor regalo que nos podriais haber dado en esta vida.

No nos podemos olvidar en ningún momento de nuestro directores de proyecto Luis Fernando Llana Díaz y Sonia Estévez Martín. Por habernos dedicado vuestro tiempo y por la confianza depositada en nosotros.

Y por último, queremos agradecer a todos nuestros amigos, aquellos que nos han acompañado durante todo este tiempo en la carrera y los que hemos conocido a lo largo de este maravilloso viaje.

Resumen

En la actualidad cada vez son más las personas que realizan deporte de forma diaria ya sea por salud, placer o necesidad. Siempre ha sido imprescindible mantener una vida sana y equilibrada además de los innumerables beneficios físicos y psicológicos que nos presta una vida activa. Pese a la gran cantidad de ventajas, que como bien decimos, conlleva la práctica de cualquier deporte, también es necesario conocer y hacer hincapié en la metodología con la que se practica, ya que realizar cualquier actividad de forma inapropiada puede acarrear graves consecuencias y problemas para la salud.

Por eso en este proyecto nos vamos a centrar en el análisis de la postura del ciclista, el cual por no considerarse un deporte de impacto o agresivo, suele pensarse que es difícil lesionarse, sin embargo, esto no es cierto puesto que se realizan movimientos cíclicos de larga duración y esto sí que puede producir lesiones en el ciclista si se están obrando de mala manera.

Finalmente, el objetivo principal de este trabajo ha sido realizar el análisis biomecánico en una aplicación móvil, ya que hasta el momento hay poca disponibilidad en el mercado de herramientas portátiles que se encarguen de realizar un análisis en cualquier momento y lugar. Queremos que ayude a perfeccionar una metodología de bajo coste accesible a todos aquellos profesionales, deportistas aficionados o personas que por necesidad, pretendan llevar a cabo la evaluación y análisis simultáneo de la flexo-extensión de las articulaciones del tobillo, la rodilla y la cadera mientras el ciclista pedalea, con la posibilidad de que posteriormente el usuario pueda conocer los movimientos que está realizando mal y en qué momento se han realizado.

Palabras clave

- Ciclismo
- Biomecánica
- Android
- Análisis
- Vídeo
- OpenCV

Abstract

Nowadays the number of people engaging in sport be it for health, pleasure or necessity is increasing daily. In the world of today is essential to have a balanced and healthy lifestyle besides the great number of physical and psychological benefits said lifestyle brings. In spite of the great number of advantages that as we say come with practising any sport, it is also important to know and make an emphasis in the methodology of the own praxis because any activity done wrong can bring forth serious consequences and health problems.

That is why in this project we are focusing in cycling analysis, a sport which is neither aggressive nor a contact sport. This brings people the false illusion that is complicated to get injured, a false fact because the continuous cyclic movements during such a long time can produce injuries if the cyclist is doing the movements wrong.

Finally the main goal of this project has been to take the biomechanical analysis to a mobile environment because to this day an application that enables such analysis everywhere and everytime cannot be found. We want to help to polish a low cost methodology accessible to professionals, amateur athletes o people that by their necessities, want to perform and analysis and simultaneous evaluation of the flexing and extending of the ankle, knee and hip joints while the cyclist is pedaling to let them know what movements are doing and when.

Keywords

- Cycling
- Biomechanics
- Android
- Analysis
- Video
- OpenCV

Índice general

Autorización	3
Agradecimientos	v
Resumen	vii
Palabras clave	ix
Abstract	xi
Keywords	xiii
1. Introducción	3
1.1. Motivación	3
1.2. Objetivos	4
1.3. Estructura de la memoria	5
2. Introduction	7
2.1. Motivation	7
2.2. Objectives	8
2.3. Memory Structure	9
3. Estado del Arte	11
3.1. Primeras apariciones de la biomecánica	11
3.2. El problema de las lesiones en el ciclismo	12
3.3. Estudios biomecánicos en la actualidad	14
3.3.1. Estudios biomecánicos en centros especializados	15
3.3.2. Aplicaciones móviles biomecánicas del mercado	18
4. Tecnologías utilizadas y alternativas	21
4.1. Android	21
4.2. OpenCV	22

4.3. JavaCV	23
4.4. Java	23
4.5. XML	24
4.6. GraphView	25
5. Versiones Android compatibles con nuestra aplicación	27
6. Metodología de desarrollo software	29
6.1. ¿Por qué utilizar una metodología ágil?	30
7. Diseño de la aplicación	31
7.1. Mockups	31
7.2. Aplicación actual	37
8. Dificultades encontradas y sus soluciones	49
8.1. MediaRecorder: start failed: -19	49
8.2. Cabeceras nativas C++ (JNI Android Studio)	50
8.3. Motivo y razón del uso de la orientación vertical de nuestra aplicación	52
8.4. Problema de lectura del video por OpenCV	52
8.5. Importación de OpenCV y JavaCV en Android Studio	53
8.6. Problemas de rendimiento	54
9. Manual de la aplicación	55
9.1. Grabación y reproducción	55
9.2. Análisis del vídeo	56
9.2.1. Recorte del vídeo	57
9.2.2. Degradado de la imagen y generación del vídeo	58
9.3. Resultados	59
9.4. Posibles problemas	60
9.4.1. Memoria insuficiente	60
9.4.2. Permisos	61
10. Materiales y elementos necesarios para probar la aplicación	63
11. Funcionamiento interno de la aplicación	65
11.1. Grabación del video	65
11.2. Edición de imagen	65
11.3. Análisis del video original y creación del video con los ángulos . . .	66
12. Contribuciones al proyecto	71
12.1. Carlos Jaynor Márquez Torres y Laura Pérez Jambrina	72
12.2. Daniel Tocino Estrada	74

13.Líneas de trabajo futuro	77
14.Conclusiones	79
15.Conclusions	81

Capítulo 1

Introducción

1.1. Motivación

Cada vez son más las personas que utilizan la bicicleta para hacer deporte, pasear o como medio de transporte habitual, tan sólo es necesario mirar alrededor en cualquier calle para darse cuenta la facilidad que tenemos para poder alquilar una bicicleta urbana o ver como muchas personas practican deporte en ella, siendo un hecho imparable que no se atiene a una moda y cuyos beneficios son patentes tanto para el medio ambiente, como para la salud ya que se recomienda como excelente ejercicio cardiovascular.

Pero la verdad es que la realización de este tipo de deporte puede provocar también numerosas lesiones [LESa], como condromalacia rotuliana [CON], lesiones de columna y lesiones de los miembros inferiores y superiores, debido a que el ser humano no está diseñado para montar en bicicleta, porque nuestro sistema esquelético-muscular está diseñado para realizar actividades como andar, correr... Pero no hemos evolucionado para apoyar la base de la columna vertebral en una pequeña superficie como es el sillín; ni para realizar actividades durante un largo periodo tiempo que impliquen tener el tronco inclinado hacia delante, ni sosteniendo durante varias horas nuestro peso con las manos, sumándole además las velocidades que se pueden llegar a alcanzar y actuando sobre diversas superficies de rodaje. Por este motivo, es de esperar que aparezcan lesiones, molestias y dolores cuando pretendemos pasar muchas horas realizando una actividad para la que no estamos realmente preparados desde un punto de vista evolutivo.

Por ello podemos afirmar que cada ciclista necesita una adaptación [MEDa] personalizada de su bicicleta y debe de tener en cuenta que aunque sea de su talla no quiere decir que esté perfectamente adaptada a ella, y que es imprescindible ajustar con atención la altura, inclinación, retroceso y ángulo del sillín y manillar. Así que para conseguir una correcta postura lo mejor es hacer un estudio biomecánico [IMPb] para adecuar las métricas de la bicicleta en base a las potencialidades o limitaciones del ciclista, buscando mayores niveles de confort, rendimiento y evitar lesiones.

Por todo ello decidimos lidiar con estas necesidades de los usuarios desarrollando una aplicación móvil, que llamamos *AngleBike*, para así que de una manera fácil y sencilla, pudieran utilizarla cuando quisieran para analizar los movimientos que realizan al montar en bicicleta y así poder posteriormente mediante un análisis adoptar una postura adecuada encima de cualquier bicicleta y poder darse cuenta de ciertas posturas inadecuadas que se toman.

1.2. Objetivos

El objetivo principal de *AngleBike*, es realizar un análisis biomecánico de la posición del ciclista mientras pedalea para que así pueda optimizar su posición y lograr un ajuste perfecto.

Además podemos llegar al alcance de cualquier persona, que quiera comprobar que está realizando su actividad en una buena posición y que tienen un patrón motor adecuado, de usuarios que quieren optimizar su rendimiento ó de aquellos usuarios que tienen dolencias y no son capaces de disfrutar de su actividad. Teniendo por lo tanto en consideración, a personas que utilizan la bicicleta como tema deportivo, hobby ó por necesidad de transporte. Logrando así un estudio biomecánico de forma económica, alcance de todos y en cualquier momento.

A continuación, presentaremos los objetivos que nos propusimos alcanzar:

- Aprender a desarrollar aplicaciones móviles, mediante el uso de Android Studio.
- Aprender a utilizar la biblioteca OpenCV para lograr procesamiento de imágenes y visión computarizada.

- Comprender y lograr una funcionalidad correcta de Android con OpenCV.
- Desarrollar el estudio biomecánico en una aplicación para dispositivos móviles.
- Desarrollar gráficos que informen al usuario de los ángulos que está adoptando a lo largo de su actividad y que pueda interactuar con ellos en tiempo real para saber en qué momento obtuvo dicha postura y posición.
- Ofrecer la posibilidad de realizar un estudio biomecánico accesible a todo el mundo, cuya realización se pueda hacer en cualquier momento y sin coste alguno, cosa a tener en cuenta debido a que dichos estudios hoy en día, están en pleno auge y son excesivamente caros.
- Evaluar el uso de la aplicación y su aceptación con usuarios reales.

1.3. Estructura de la memoria

La estructura que hemos seguido para organizar esta memoria consta de los siguientes capítulos.

Primeramente nos encontraremos con los capítulos 1 y 2 redactados en español e inglés respectivamente, en los que se introduce el contexto en el que se realiza este proyecto, además de la motivación y los objetivos que se llevan a cabo. Por último, se explica la estructura de este documento.

En el Capítulo 3 se describe las primeras apariciones de la biomecánica y los problemas que padecen los ciclistas debido a las malas posturas que adoptan en la bicicleta. Y en segundo lugar una descripción tanto de centros físicos como de aplicaciones móviles que realizan estudios biomecánicos desde menos envergadura hasta mayor escala.

En el Capítulo 4 se describen todas las tecnologías que hemos tenido que utilizar para la realización de nuestro proyecto.

En el Capítulo 5 se detalla y se justifica qué versiones de Android son totalmente compatibles con nuestra aplicación.

En el Capítulo 6 se explica la metodología de desarrollo software que hemos utilizado para que nos ayudara a estructurar, planificar y controlar el proceso de

desarrollo de nuestro proyecto y la justificación de la elección de esa metodología.

En el Capítulo 7 se muestra y se explican los pasos por los que ha ido pasando el desarrollo gráfico de nuestra aplicación, desde los primeros prototipos que diseñamos para la interfaz gráfica como los que finalmente podéis encontrar en nuestra aplicación móvil, juntos con una justificación de dichos cambios.

En el Capítulo 8 se redactan las dificultades que nos hemos ido encontrando a lo largo de todo el proceso de creación de la aplicación, además se explican cómo hemos logrado solucionar dichos problemas.

En el Capítulo 9 se presenta el funcionamiento que tendrá la aplicación, para facilitar la experiencia del usuario y que no haya confusión durante la ejecución de la misma.

En el Capítulo 10 se presentan los materiales y herramientas necesarias para poder probar la aplicación.

En el Capítulo 11 se habla sobre cómo se realiza internamente el análisis y estudio de los movimientos del ciclista, para posteriormente poderle mostrar al usuario tanto la gráfica de sus ángulos obtenidos en el tiempo como el vídeo final, donde aparecen dibujados los ángulos logrados.

En el Capítulo 12 se comentan las aportaciones que hemos realizado al proyecto cada uno de los autores del trabajo.

En el Capítulo 13 se detallan las futuras líneas de trabajo de este proyecto.

En los Capítulos 14 y 15 redactados en español e inglés respectivamente, se analizan las conclusiones que hemos obtenido después de la finalización de nuestro proyecto.

Capítulo 2

Introduction

2.1. Motivation

Everyday there is more people using a bicycle to exercise, take a stroll or as a transport. It is clear just by looking around how easy is to rent an urban bike or catch people using one, it is not only a trend but an way of living that will continue to grow, which benefits extend to the environment as well as one's health because cycling is a great cardiovascular exercise.

But the truth is that doing this kind of sport can also carry some injuries [LESa] like chondromalacia patellae [CON], “*runner's knee*”, as well as injuries in the spine and the lower and upper limbs because humans are not designed to ride a bike. Our skeleton and muscular system is designed for tasks such as walk and run, we have not evolved in a way to rest the base of the spine in a small area like the saddle nor to stay for long periods of time with our trunk inclined forward or using our hands to carry our weight for hours, adding to this the great velocities we can reach and the friction it brings. That is why is to be expected for disturbances, pains and injuries to surface when we plan to invest a great deal of hours to an activity not fit for us from an evolutionary standpoint.

For the reasons stated above we can assert that each cyclist needs a personalized adaptation [MEDa] to his bicycle and should consider that even if the bike is his size that does not mean that is perfectly prepared for him and that is of the utmost importance to adjust the height, inclination, recoil and angle of the saddle and handlebar. So, in order to get a correct posture, the best thing is to take

part in a biomechanical study [IMPb] to adjust the bicycle's metrics to the biker's potency and limitations seeking great levels of comfort, performance and to avoid injury.

That is why we decided to tackle these necessities of the users by developing a mobile application, that we decided to call *AngleBike*, to give them an easy and simple way to analyze the movements they do while biking and later via an analysis allow them to reach a correct posture in any bicycle and realize the bad postures they take.

2.2. Objectives

The main objective of *AngleBike* is to do a biomechanical analysis of a cyclist position while he pedals so he can optimize his position and reach a perfect adjustment.

We also can reach any person that wants to check if it is doing the activity in a good posture and has an adequate motor pattern, users that want to optimize their performance or users that are having pains and cannot enjoy their activity. We consider then people that use the bicycle as a sport, hobby or transportation need, reaching a cheap biomechanical study available to all at any moment.

Here are the objectives we aimed to reach:

- Learn to develop mobile applications by using Android Studio.
- Learn to use the OpenCV library to get image processing and computer vision.
- Learn how to make a connection between both of them.
- Develop a biomechanical study in a mobile application.
- Develop graphics that notify the user of the angles adopted while he was exercising and reach interactivity with them to know in real time in what moment he got that posture and position.
- Bring the possibility of doing a biomechanical study accessible to all whose attainment can be done anytime and with zero cost, a thing to take in account because said studies are going up in popularity and are expensive.

- Evaluate the application use and acceptance with real users.

2.3. Memory Structure

The structure we have followed to organize this memory consists of the following chapters:

First, we find chapters 1 and 2 redacted in both Spanish and English respectively, in which we introduce the context that surrounds this project and the motivation and objectives that are carried out. At last we explain this document's structure.

In chapter 3 at first, we describe the first appearances of biomechanics and the troubles cyclists face from bad posturing. Secondly, we describe centers and mobile applications that do biomechanics studies from low to high scope.

At chapter 4 we find all the technologies used in the development of this project.

On chapter 5 we detail and justify what android versions are compatible with our application.

In chapter 6 the methodology of software development we have followed is explained, which helped us structure, plan and control the development process of our project and the justification of that decision.

Chapter 7 shows and explains the steps that our graphic development has followed, from the first prototypes we designed to the end results along the justification of the changes.

On chapter 8 are written the difficulties we have faced along the full process of development. Also, we explain how we overcome such problems.

Chapter 9 presents the behavior our application has, to ease user experience and to avoid confusion during execution.

Chapter 10 presents the materials and tools needed to enable you to apply.

At chapter 11 we talk about how the internal process of the cyclist analysis works so later we can offer the user a graph of the angles made during that time and the

video result where those angles are drawn.

In chapter 12 each member of the group explains their contributions to the project.

On chapter 13 we talk about future lines of work.

Chapter 14 and 15, written in both Spanish and English, analyzes the conclusions extracted from our project.

Capítulo 3

Estado del Arte

Para la realización de la aplicación AngleBike, hemos llevado a cabo un estudio sobre la necesidad y demanda actual de estudios biomecánicos y las principales funcionalidades que poseen algunas de las aplicaciones existentes en este ámbito, detectando sus carencias y sirviendo estas de justificación para la creación final de nuestro prototipo.

3.1. Primeras apariciones de la biomecánica

En este apartado queremos exponer, la historia de cómo surgió la biomecánica deportiva y quienes fueron sus pioneros, para hacer comprender al lector la importancia que ha tenido y sigue teniendo en la actualidad y el porqué se siguen haciendo numerosos estudios.

Primeramente vamos a definir la biomecánica [BIO] como el estudio de la aplicación de las leyes de la mecánica a la estructura y al movimiento de los seres vivos con el fin de conseguir que esas acciones reiteradas sean lo más eficaces y lo menos dañinas posibles para el ser humano.

Ya en los tiempos de la Grecia clásica, cuando se empezaron a jugar los juegos olímpicos [JUE], se tiene constancia de que el científico Aristóteles fue uno de los fundadores de la biomecánica ya que realizaba múltiples observaciones descriptivas de los movimientos de los animales. Pero el principal pionero de la biomecánica

tal y como la conocemos hoy en día, fue Leonardo Da Vinci ya que fue quien tomó nota de los primeros trabajos de Aristóteles sobre las leyes de movimientos en animales, y los perfeccionaría también en humanos.

A medida que a lo largo de la historia el deporte y el entrenamiento diario regresaban a un plano de mayor importancia, provocarían el comienzo en el siglo XIX de nuevos estudios, esta vez protagonizados por el anatomista ruso P.F. Lesgaft [LESb], al cual se le considera el padre de la biomecánica deportiva moderna. El deporte se hizo tan imprescindible y competitivo que las disciplinas olímpicas se tomaría como una causa nacional en la Unión Soviética [UNI], quienes comenzarían un largo programa de perfeccionamiento de sus gimnastas siguiendo el modelo del libro de Lesgaft, *“Biomecánica de los Ejercicios Físicos”*, publicado en 1939, para mejorar el gesto motor del cuerpo realizado en las diferentes disciplinas deportivas y así evitar lesiones. Todo esto provocó que en todos los institutos soviéticos de cultura física, la biomecánica se convirtiera en una asignatura obligatoria; además, se crearon cátedras de biomecánica, se elaboraron programas, se editaron manuales y libros de texto. Cuyo fin era sistematizar los conocimientos durante la formación y el perfeccionamiento de la disciplina docente, lo que ejerció influencia sobre el desarrollo de la teoría de la biomecánica en el deporte.

En la actualidad, se ha continuado realizando estudios biomecánicos especializados y muchos de ellos centrados ya en disciplinas deportivas concretas con el fin de seguir optimizando el rendimientos de las personas, para desarrollar un sistema de entrenamiento adecuado, óptimo y evitar el riesgo de lesiones a la hora de realizar cualquier tipo de deporte.

3.2. El problema de las lesiones en el ciclismo

Todas las modalidades deportivas causan una serie de lesiones, sobre todo si se practica dicha modalidad de forma reiterada. En nuestro caso, la disciplina del ciclismo, se le atribuyen distintos tipos de lesiones, las cuales se producen fundamentalmente desde dos aspectos a tener en cuenta:

- Los errores en la planificación y programación de la actividad.
- El gesto deportivo reiterado y basado en alteraciones morfológicas o biomecánicas.

Podemos encontrar personas como el especialista en Medicina del Deporte, José Ramón Gómez Puerto que afirma [Pue08] que *“Las consultas médicas más frecuentes del ciclista suelen ser por cervicodorsalgias y problemas del aparato extensor de la rodilla. El diagnóstico incluye la valoración del dolor, la exploración física, los factores constitucionales y los estudios por imagen. En los ciclistas, además, el examen de la bicicleta y del calzado son muy recomendables. Las formas de prevención pasan por modificar el gesto deportivo y cumplir con las reglas básicas para una posición correcta sobre la bicicleta. Es determinante la medición de la altura perineal, independientemente de la talla, para elegir correctamente la bicicleta.”*

En definitiva, cualquier alteración en el acoplamiento con la bicicleta puede generar, a nivel de los ligamentos y de los apoyos, tensiones y dolores anormales que constituyen posibles fuentes de inflamación. Por lo que, las posturas incorrectas sobre la bicicleta, mantenidas durante mucho tiempo son capaces de alterar el gesto deportivo normal y producir patologías que será necesario diagnosticar y tratar correctamente, además de empeorar la rapidez y eficacia del ciclista [DdT97]

A continuación podéis ver en la tabla mostrada en la Fig. 3.2 algunas consecuencias que poseen algunos defectos [TAB], tanto referidos a la bicicleta como puede ser la altura y el alcance del cuadro de la bicicleta 3.1, como al pedaleo.

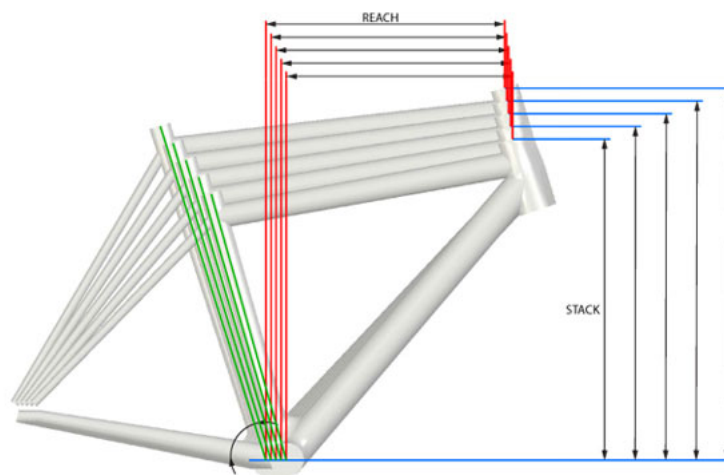


Figura 3.1: Diferentes alturas y alcances del cuadro de una bicicleta

Defecto	Consecuencia
Cuadro de la bicicleta largo	Obliga al ciclista a estar una posición más horizontal.
Cuadro de la bicicleta corto	Obliga al ciclista a adoptar una posición más vertical, pudiendo genera lumbalgias
Cuadro de la bicicleta alto	Crea una hiperextensión de las rodillas lo que produce una sobrecarga en la musculatura posterior
Cuadro de la bicicleta bajo	Provoca una sobrecarga de la musculatura extensora
Excesiva rotación interna del pie	Sobrecarga de la musculatura externa y bíceps
Excesiva rotación externa del pie	Provoca sobrecarga de los tendones de la región interna de la rodilla y tendones de la pata de ganso
Mal alineamiento de la rótula	Dolores en la rótula, provocando condromalacia rotuliana
Altura y posición incorrecta del sillín	Dolor lumbar
Sillín muy atrás y muy alto	Provoca dolor en la rótula

3.3. Estudios biomecánicos en la actualidad

Como hemos comentado en los apartados anteriores 3.1 y 3.2, la realización de un estudio biomecánico para la modalidad deportiva como es el ciclismo es muy importante. Por ello, antes de comenzar nuestro proyecto decidimos investigar sobre los estudios que se están realizando en la actualidad, ver en qué consisten y qué pruebas se realizaban en ellos, también es lícito y de gran importancia resaltar, que nosotros íbamos a realizar una aplicación móvil, por lo que decidimos explorar y examinar dos campos totalmente diferenciados:

1. Estudios biomecánicos en centros especializados para ello. Apartado 3.3.1
2. Aplicaciones móviles que se centraran en un objetivo igual o parecido al nuestro.

A continuación en los siguientes apartados os contaremos, las aplicaciones y centros encontrados que se adaptan y tenían relación con el objetivo de nuestro proyecto.

3.3.1. Estudios biomecánicos en centros especializados

Nuestra finalidad, era conocer realmente qué servicios se suministraban en centros físicos, siempre centrándonos en estudios biomecánicos aplicados solamente a la actividad deportiva del ciclismo.

Entre los centros encontrados, hemos decidido mostrar los más representativos:

- Custom4.us



Figura 3.2: Logo de la compañía Custom4

Este centro se dedica en vez de examinar o arreglar la métrica de la bicicleta, su propuesta [CUS] es solucionar desde raíz el origen preciso del dolor que se le manifiesta al deportista, por lo que no solo examinaban la bicicleta, sino también al ciclista, explicando en qué falla su técnica de pedaleo o de posicionamiento y modificándolo, ofreciendo tres servicios diferentes destinados a perfiles de ciclistas distintos:

- Básico: Destinado a ciclistas amateurs que no presentarán ningún tipo de molestia o lesión y que simplemente quisieran obtener un ajuste correcto de sus bicicletas y calas. Ofreciendo el ajuste del sillín, del manillar, de las calas y del baricentro. Con un precio de 50 €.
- Avanzado: Abocado a ciclistas que exponen molestias y lesiones de rodillas, cuello, muñecas y pies. Brindando el ajuste del sillín, del manillar, de las calas y del baricentro y realizando valoraciones anatómicas del sillín, de las disimetrías de cadera. Con un precio de 150 €.
- Elite: Expuesto a ciclistas con lesiones graves, como pueden ser protusiones discuales y disimetrías severas. Y también ciclistas de alta competición que quieren obtener un mayor rendimiento. A parte de ofrecer

los servicios de los anteriores estudios, era el único que ofrecía el análisis cinético de la pedalada, con el que se podría observar dónde y en qué momento del ciclo de pedaleo se ejercía mayor presión y fuerza y la medición de ángulos para la correcta colocación de los diferentes componentes mecánicos. Por un precio de 250 €.

- EnBici



Figura 3.3: Logo del taller EnBici

Es un taller donde se realiza el Estudio Biomecánico en dos planos [ENB], por un lado se tiene en cuenta la bicicleta, y por otro los movimientos de la persona encima de la bicicleta. Para así analizar el movimiento característico de cada ciclista y configurar su bicicleta para adaptarse mejor a sus necesidades y capacidades.

Lo que consiguen estudiando ambos planos es poder modificar las métricas y también corregir ciertas costumbres, como pueden ser malas posiciones o movimientos poco eficientes del usuario cuando realiza este tipo de deporte.

Este taller consta de tres metodologías distintas, optando por estudios de menos a más precisión, los cuales son:

- Starter: En el cual se toman las mediciones, tanto del usuario, para así aconsejarle a la hora de la compra de una bicicleta, para que está se intente adaptar de la mejor forma posible a su fisiología y también se realiza una plantilla, para que el cliente pueda tomar la decisión correcta a la hora de proceder a su compra. Cuyo precio es de 50€.
- Advanced: En el que se realiza la toma de medidas, la configuración de la bicicleta al usuario y el ajuste de las calas para conseguir una óptima posición. Con un precio de 90€.
- Professional: Es la única metodología en la que se realiza la medición 3D, para adaptar la bicicleta a la persona. Su precio no aparece disponible en la web.

- TRIMAD

Es una tienda que se dedica también a realizar estudios biomecánicos [TRI], la cual incluye una novedad en diferencia con otros centros visto hasta ahora y es que trabaja con Retül [RET], qué es el sistema de análisis biomecánico



Figura 3.4: Logo de la tienda TRIMAD

que permite la medición de ángulos mientras el ciclista pedalea, consta de un arnés con unos sensores led infrarrojos que se colocan en las articulaciones claves involucradas en el movimiento reiterativo del ciclista y también consta de un escáner que analiza la luz infrarroja emitida por los sensores led (Como se puede ver en la figura 3.6), transmitiendo esta información a un ordenador y mostrando una información exhaustiva y muy precisa de cómo el usuario está posicionado encima de la bicicleta. Después de haber obtenido dichos ángulos que el ciclista ha obtenido realizando el ejercicio, lo que se intenta es acercarse a unos parámetros que provocarán una eficiencia y comodidad en el ciclista sobre su bicicleta.

La ventaja principal de este sistema es que debido a que la medición de ángulos se realiza mientras la persona pedalea, se puede obtener una visión más realista del conjunto, ya que no es lo mismo realizar un estudio quieto encima de la bicicleta, que totalmente en movimiento.



Figura 3.5: Logo del sistema Retül

En este lugar se dan a escoger tres opciones de análisis distintas:

- BikeFit: el cual incluye estudio biomecánico básico para posicionamiento correcto en la bicicleta, el ajuste básico y manual de calas y la entrega de un informe, con un precio final de 80€.
- FindBike: la finalidad de esta opción es realizar un estudio biomecánico que te ayuda a la hora de averiguar la geometría, marca, modelo y talla que más te conviene en el momento de comprarte una bicicleta. Cuyo precio es de 150€.
- Retül Bike Fitting: es la única modalidad que incluye el estudio biomecánico con la ayuda del sistema Retül, cuyo uso se ha explicado anteriormente 3.3.1. El precio de este estudio es de 150€.



Figura 3.6: Utensilios utilizados en el sistema de análisis biomecánico Retül

Después de haber investigado y haber encontrado tanta variedad de centros en los que se realiza este tipo de estudios, podemos ver que los precios son bastante elevados, por eso dichas cantidades se han expuesto en la redacción de las modalidades encontradas.

Así que nuestras conclusiones es que la aplicación que llevaremos a cabo se adapta y se parece bastante a lo que utilizan muchos sitios en la actualidad, ya que nosotros realizaremos también la medición de ángulos mientras que se producen los movimientos reiterados del ciclista y después él podrá comparar esas mediciones a una serie de parámetros, con los cuales pueda obtener una correcta postura. Y contamos con otra ventaja a tener en cuenta y es que la aplicación se encontrará alcance de todo el mundo, ya que no tendría un precio elevado como los observados anteriormente, solo tendrían que comprar los leds para el análisis.

3.3.2. Aplicaciones móviles biomecánicas del mercado

Debido a que nuestra aplicación hace un estudio biomecánico sobre el ciclista, hemos decidido buscar información sobre las distintas aplicaciones que hay en el mercado cuya función sea parecida o se acerque a lo que hace nuestra aplicación.

3.3.2.1. Strava

Es una de las aplicaciones más conocidas para hacer deporte en general [STR], no solo para ciclismo, como se puede ver en la imagen 3.7, la principal funcionalidad es que graba los tramos que se recorre para poder llevar un seguimiento del rendimiento y la mejora que se obtiene, otra posibilidad que te permite es que puedes ver y comparar los resultados con otros usuarios que han realizado esa ruta.

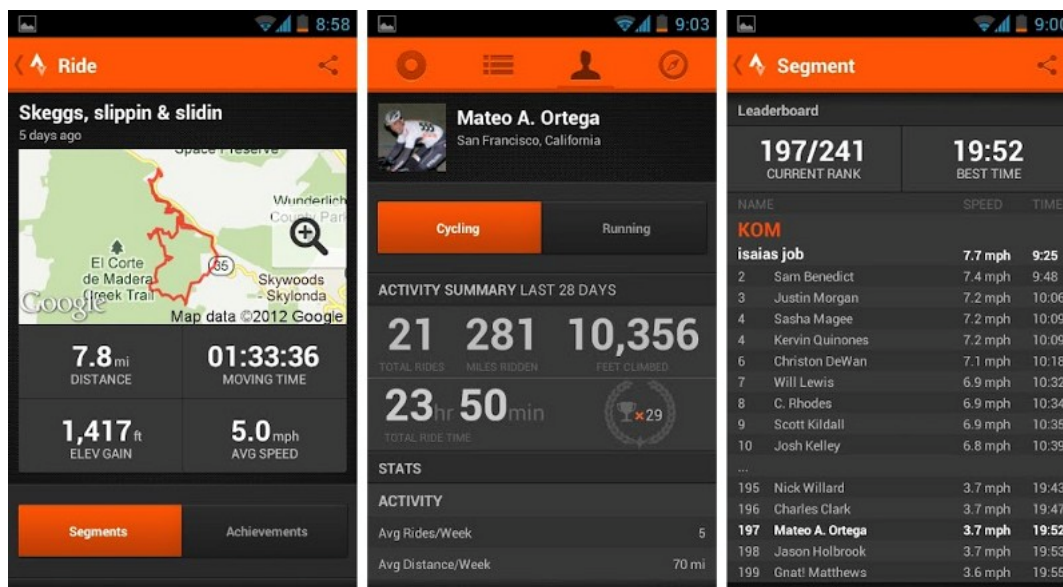


Figura 3.7: Aplicación Strava

3.3.2.2. Bike Fit

La funcionalidad principal de esta aplicación es encontrar y proporcionar la medida de la bicicleta que más se ajuste al ciclista [BIK], para ello toma una serie de datos como la medida de la persona y el terreno sobre el que se va a practicar el deporte, si es carretera o montaña, con estos valores la aplicación proporciona una imagen con las medidas aproximadas que debe tener el cuadro de la bicicleta.

A continuación podemos ver en la Figura 3.8, los tamaños del cuadro de la bicicleta y la altura y retroceso del sillín qué debe de adoptar la bicicleta del ciclista, gracias a la inserción de datos introducidas anteriormente por el usuario.

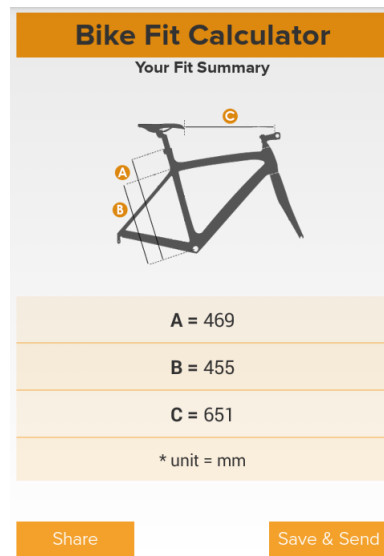


Figura 3.8: Resultado final de la aplicación Bike Fit

3.3.2.3. Fitreme. bike fitting app

La funcionalidad de esta aplicación es la que más se acerca a lo que intentamos buscar con la nuestra. Utilizando la cámara se graba al usuario en una bicicleta estática, para posteriormente en el reproductor de la aplicación el usuario pueda manipular un ángulo que aparece dibujado con el objetivo de medir la rodilla, pero es necesario ir parando el video [FIT].

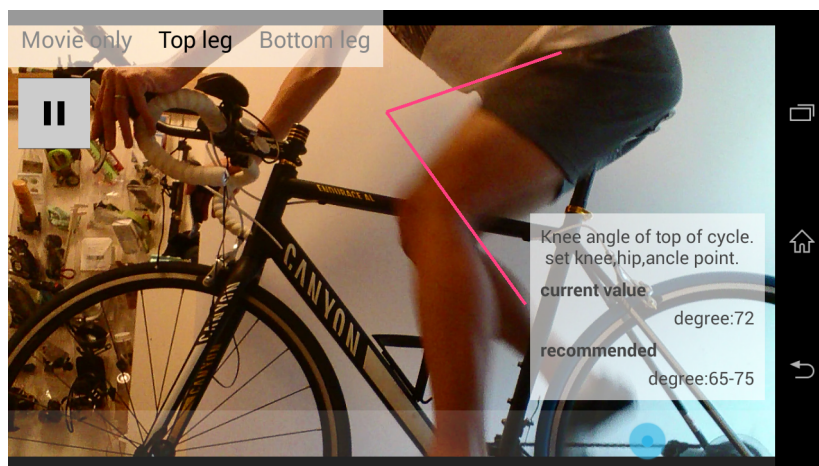


Figura 3.9: Aplicación Fitreme. bike fitting app

Capítulo 4

Tecnologías utilizadas y alternativas

En este apartado describimos las tecnologías utilizadas que nos han servido para llevar a cabo la implementación de nuestra aplicación móvil.

4.1. Android

Es un sistema operativo basado en Linux y lanzado oficialmente en 2008 por Andy Rubin, este abarca varios tipos de mercados como son los teléfonos inteligentes, Tablets, Android TV, Android Wear y Android auto. Es de código abierto ya que Google liberó Android bajo licencia Apache, lo que lo convierte en un sistema operativo totalmente libre para que un desarrollador no solo pueda modificar su código sino también mejorarlo.

Además Android permite instalar lo que quiera al propietario de un terminal, provocando una libertad plena para que desarrolladores o empresas puedan realizar aplicaciones ajustadas a sus propias necesidades.

Las aplicaciones se desarrollan en lenguaje Java aunque también se pueden incluir llamadas a cabeceras nativas con librerías como OpenCV.



Figura 4.1: Logo de Android

La otra opción disponible era desarrollar para dispositivos de Apple, ya que también tienen compatibilidad con la librería de procesamiento que hemos usado. El por qué de desechar esta opción venía dado por varios factores:

- La familiaridad con el entorno, ya que Android está basado en Java y todo los integrantes tenemos experiencia con ese lenguaje, no así con el lenguaje Swift utilizado en desarrollo iOS.
- El coste de la licencia de programador de esta plataforma, que en Android no es necesaria.
- El más importante, ninguno de los miembros contábamos con dispositivos de la compañía para poder desarrollar y hacer pruebas.

4.2. OpenCV

Es una librería software open-source de visión artificial y machine learning contando con más de 2500 algoritmos de este tipo, cuyo uso permite el reconocimiento de imágenes y vídeos tanto de objetos como reconocimiento facial y el tratamiento de estos.

Cuenta con una licencia BSD, es una librería muy usada a nivel comercial, donde podemos encontrar compañías como Google, Yahoo, Microsoft, Intel, IBM, Sony y Toyota. Todo esto se debe gracias a que es una librería multiplataforma, lo

que favorece su utilización tanto en sistemas operativos como GNU/Linux, Mac OS, Windows como en teléfonos inteligentes. Y además OpenCV está escrito en C++, y se puede programar en C, Python, Java y MATLAB.

Sus principales aplicaciones son: la detección de intrusos en vídeos, monitorización de equipamientos, ayuda a navegación de robots e inspeccionar etiquetas en productos.



Figura 4.2: Logo de OpenCV

4.3. JavaCV

JavaCV [JAVb] es un wrapper en lenguaje Java que engloba algunas funciones de OpenCV junto con otras librerías usadas en el ámbito de la visión computacional, es un proyecto independiente no oficial, cuyo código se encuentra en GitHub y se va actualizando poco a poco añadiendo funcionalidad.

4.4. Java

Es un lenguaje de programación orientado a objetos [JAVa] y la primera plataforma informática creada por Sun Microsystem en 1995.

Lo hemos utilizado para realizar la lógica de nuestra aplicación móvil. Si bien es cierto que el lenguaje Java que se usa para Android no es exactamente el mismo que con el que estamos acostumbrados a trabajar, se diferencia muy poco del Java original ya que se utiliza la misma sintaxis y semántica, pero no incorpora en su totalidad las bibliotecas de clases de Java y APIs (Application Programming Interface) que acompañan a Java SE o ME.



Figura 4.3: Logo de Java

4.5. XML

Son las siglas del Lenguaje de Etiquetado Extensible, cuya expresión se forma a partir del acrónimo de la expresión inglesa eXtensible Markup Language y se trata de un metalenguaje extensible de etiquetas que fue desarrollado por el World Wide Web Consortium (W3C), una sociedad mercantil internacional que elabora recomendaciones para la World Wide Web.

Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.

Este lenguaje está presente a lo largo de todo nuestro proyecto, teniendo un peso fundamental dentro de él, ya que es con el que creamos las interfaces para el usuario, lo que permite la separación del diseño realizado con este lenguaje de los datos y la lógica de la aplicación hechos con el lenguaje Java, quedando dividida cada parte en sus respectivos documentos.

En nuestra aplicación se pueden encontrar archivos realizados con este lenguaje para dos motivos principales, el primero es para la correcta creación de las interfaces para el usuario, definiendo mediante él los archivos para la creación

de las vistas junto a la definición de estilos, tipografía y colores. Pero también juega un papel fundamental en la realización de un archivo llamado *AndroidManifest.xml* [MAN], el cual se encuentra situado siempre en el directorio raíz de un proyecto Android, proporcionando información esencial sobre la aplicación al sistema, la cual se necesita para poder ejecutar el código de la aplicación. En él se describen los componentes, las diferentes actividades existentes, los servicios, los receptores de mensajes y se declaran los permisos que debe tener la aplicación para acceder a las partes protegidas de una API e interactuar con otras aplicaciones, junto con la de declaración de los permisos que otros deben tener para interactuar con los componentes de la propia aplicación.

4.6. GraphView

Es una librería libre [GRAb] para Android que permite la fácil realización de gráficas. La hemos usado en nuestro proyecto para mostrar el análisis de los ángulos que ha formado el usuario en los distintos vídeos.

Capítulo 5

Versiones Android compatibles con nuestra aplicación

Nuestra aplicación es compatible con todas las versiones de Android superiores a la versión Lollipop 5.0 . Esta decisión fue tomada a lo largo que íbamos desarrollando nuestra aplicación, debido a que nos fuimos dando cuenta que muchas de las decisiones de estilos que queríamos implementar, algunas de las cuales eran además imprescindibles para nuestra aplicación, no eran compatibles con la versión de KitKat, que era la opción pensada en un principio por la gran cantidad de usuarios que podrían disponer de ella.

De acuerdo con lo comentado, vamos a exponer por consiguiente cuáles fueron los principales inconvenientes que nos hicieron tomar dicha decisión:

Ya que nuestra aplicación realiza videos, queríamos aprovechar toda la pantalla del dispositivo para mostrar la imagen completa que capta la cámara, esto nos implicaba poner transparente el componente de navegación inferior llamado “*Navigation Bar*” [NAV] para así ganar más pantalla. Existía una opción parecida en la versión KitKat pero no daba el mismo resultado, ya que lo único que hacía es ponerlo translúcido, también queríamos ocultar la barra de estado superior, donde se muestra la hora, el nivel de batería y de cobertura, llamada “*Status Bar*” [STA], cuya opción de ocultación no nos la proporcionaba KitKat [OCU].

Nuestra aplicación podría utilizarse para más de la mitad de los móviles Android que hoy en día se encuentran en el mercado, ya que las versiones Lollipop, Marshmallow y Nougat se encuentran integradas en un 66,6 % [VER] de los smartphones

como se puede ver en la imagen 5.1. Siendo además Android Lollipop la más utilizada [MAS], por encima de Marshmallow.

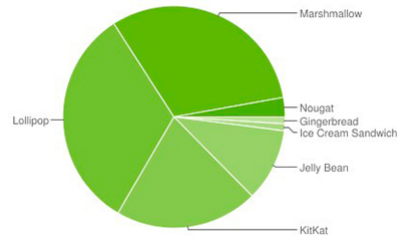


Figura 5.1: Veriones más utilizadas

Una de las ventajas de que la aplicación estuviera disponible para su utilización a partir de la versión Lollipop 5.0, es la aparición de Material design [MAT] la cual solo se integró en Android Lollipop como reemplazo de Holo [HOL]. Es una normativa de diseño enfocado en la visualización del sistema operativo Android, y además multiplataforma, la cual nos ha ayudado a la hora de realizar el diseño de nuestra aplicación.

Por último y no por ello menos importante, teníamos que tener en consideración que no todos los móviles pueden soportar un alto procesamiento de imagen. Así que nos hemos guiado por las versiones de Android más recientes, la cuales conllevarían estar implementadas en móviles más actuales en los que la duración del procesamiento se podría reducir.

Capítulo 6

Metodología de desarrollo software

Muchos de los problemas [PROb] que suceden en los proyectos y trabajos en equipo son debidos a una mala especificación de los requisitos, cambios de los requisitos funcionales, mala organización del equipo, mal reparto de trabajo y poca implicación de todas las partes del proyecto.

Todos estos problemas y circunstancias se pueden resolver con las metodologías ágiles [MET], priorizando los individuos y las interacciones sobre los procesos y las herramientas, realizando entregas parciales completamente funcionales como las que hemos tenido que hacer a lo largo de la realización de nuestro proyecto, promoviendo una colaboración activa entre todas las partes implicadas, debido a que somos tres miembros del grupo y todo tenemos que colaborar activamente y proporcionando una mayor flexibilidad frente a los cambios de requisitos.

Para la realización del proyecto se ha seguido una metodología ágil. Los motivos por los cuales se ha elegido utilizar este tipo de metodología se detallan en la siguiente sección.

6.1. ¿Por qué utilizar una metodología ágil?

El principal motivo por el cual se ha decidido utilizar este tipo de metodología es por el tipo de proyecto que hemos llevado a cabo. Ya que se trata de una aplicación móvil que tiene que realizar un análisis biomecánico de los movimientos del ciclista, por lo que, los requisitos o la forma de llevarlo a cabo podría sufrir variaciones significativas, ya que el análisis, el diseño y la construcción no son predecibles desde el punto de vista de la planificación, lo que sería realmente lo deseable. Además, de ser precavidos y estar preparados para reaccionar rápidamente ante un cambio en el proyecto, ya que podíamos comenzar con el desarrollo de alguna parte y darnos cuenta al cabo de un tiempo que a lo mejor no íbamos por un buen camino.

Además cabe destacar que, como en nuestro proyecto somos tres participantes esto propiciaba aplicar este tipo de metodología, para lograr un equipo de trabajo autoorganizados y multidisciplinar.

El uso de este tipo de metodología nos ha aportado una serie de ventajas [VEN] y beneficios que procederemos a contaros a continuación:

- La realización de las entregas y reuniones con nuestro coordinadores de forma continua, nos proporcionaban un Feedback directo, lo que nos ayudaba a nosotros a saber si íbamos por buen camino.
- Desde un principio siempre se intentaba planificar la entregas, los objetivos y los logros que teníamos que ir consiguiendo, pero también se pueden producir a lo largo de un proyecto fallos o problemas y la entrega continuada y la retroalimentación recibida en las distintas etapas nos ayudaban también a ir logrando mediante esta metodología una planificación incremental y un proceso adaptable a cambios técnicos y de negocio para asegurar un mejor resultado.
- Producción incremental, ya que cada entrega y reunión aportaba un valor incremental en el resultado de nuestra aplicación.
- Reducción de riesgos a largo plazo, gracias a la reuniones y retroalimentación recibida.
- Conocer el progreso real del proyecto desde el inicio.
- Colaboración en equipo.

Capítulo 7

Diseño de la aplicación

En esta sección vamos a mostrar los pasos por los que ha ido pasando el desarrollo gráfico de nuestra aplicación, desde los primeros prototipos que diseñamos para la interfaz gráfica como la interfaz final que podéis encontrar en nuestra aplicación.

7.1. Mockups

Os vamos a presentar los prototipos principales que se diseñaron al principio, éstos nos fueron muy útiles porque nos sirvieron de guía para saber qué era lo que queríamos realmente lograr y ofrecer al usuario. Esto que puede parecer a simple vista una cosa poco importante nos ha sido realmente importante para aclararnos de cuáles eran nuestros objetivos y prioridades en la interfaz gráfica y en la posterior implementación lógica de ésta y además nos sirvió para mejorar nuestra organización al saber cuáles son las tareas que íbamos realizando y las que quedaban por complementar.

Todos los prototipos que vamos a presentar a continuación están realizados con la herramienta Justinmind Prototyper [JUS], la cual permite definir la estructura, funcionalidades y diseño antes de comenzar con el desarrollo en sí [PROc].

El primer mockup que realizamos fue el diseño de la vista previa de la cámara, ya que era una vista principal y primordial en nuestra aplicación. Es el punto de partida de nuestro proyecto debido a que cualquier usuario que quiera realizar su

análisis lo primero que va a tener que hacer es acceder a está vista para proceder a la grabación de su actividad, por ello decidimos ponerle el nombre de Inicio.

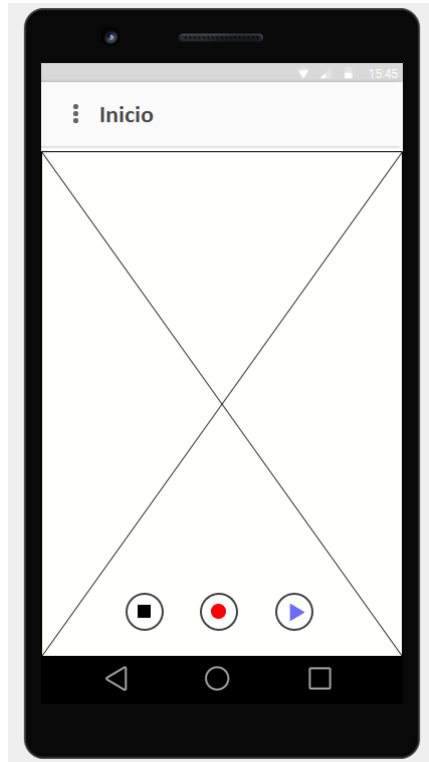


Figura 7.1: Mockup 1: Inicio

para finalizar la grabación y a su lado derecho el botón *“rec”*, que comenzaría con la grabación del vídeo. El tercer botón, menos relevante, lo diseñamos con el fin de que nada más detener la grabación realizada el usuario tuviera la opción de darle al botón del *“play”* para poder reproducir lo que acaba de terminar de grabar.

Como se puede observar también, dicha vista 7.1 cuenta con un menú en la parte superior de la pantalla, que nos indicará en todo momento en qué sección de la aplicación nos encontramos y un menú deslizante para proceder a la navegación dentro de nuestra aplicación con la finalidad de poder cambiar de escenario.

El segundo prototipo que nos pusimos a realizar fue la vista del menú, muy importante también para plasmar desde un principio cuáles eran nuestros objetivos y prioridades de la aplicación para así fijar y establecer una serie de secciones que bajo ningún concepto podrían desaparecer.

Debido a la importancia que tiene, lo fundamental para nosotros era conseguir una vista previa de la cámara que ocupará todo lo que se pudiera para así aprovechar toda la dimensión que la pantalla que nuestro móvil nos está ofreciendo. Decidimos que para maximizar el espacio disponible nuestros botones serían objetos flotantes y después de tomar esa decisión procedimos a investigar y reflexionar cuáles serían relevantes y necesarios para las personas que utilizarán nuestra aplicación, con lo que decidimos basarnos en la interfaz de la cámara [CAMb] que los propios móviles Android traen instalada de forma nativa ya que esto proporcionaría a nuestros usuarios familiaridad y comodidad a la hora de utilizar la nuestra. Por ello, como se puede observar en la figura 7.1, decidimos colocar tres botones, dos de ellos totalmente imprescindibles, los cuales serían los dos botones inferiores izquierdos, el botón de *“stop”*

Siempre tuvimos en cuenta que según fuéramos desarrollando el proyecto lo que sí qué podría suceder es la aparición de alguna funcionalidad extra que podríamos proporcionarle al usuario, pero en un principio el menú que se creó 7.2 y en el que nos basamos en todo momento fue el que se proporciona a continuación.

Este menú contaba con cuatro secciones imprescindibles:

- Inicio: En el cual podemos encontrar la vista previa de la cámara 7.1 en la que se puede proceder a realizar la grabación de la actividad del ciclista como hemos comentando anteriormente.
- Galería: Donde podríamos encontrar un listado con todos los videos 7.3 que hemos realizado de nuestras actividades para posteriormente seleccionar uno de ellos y poderlo reproducir 7.4.
- Imágenes: Está sería una sección donde poder encontrar un listado de frames 7.3 obtenidos de cada uno de los vídeos grabados anteriormente, es decir, que por cada video grabado y guardado en la sección “*Galería*” aquí podremos encontrar un frame correspondiente a cada uno de ellos, para finalmente poder seleccionar uno de los frames presentes en la lista, elegir una serie de parámetros que nos ayuden a solamente dejar visibles en nuestra imagen los leds 7.5 que lleva incorporados el ciclista en su cuerpo y proceder con esos parámetros establecidos al análisis del video completo.
- Gráficos: En esta sección nuestra intención era reflejar un listado de archivos 7.3, y qué si seleccionabas uno de ellos nos encontraríamos con un gráfico 7.5 en el cual en el eje “X” se encontraría reflejado el tiempo del video y en el eje “Y” la progresión de los ángulos alcanzados por el usuario. de forma que según el video que se haya analizado y las posturas que haya ido obteniendo el ciclista, se creará un gráfico donde se establecieran puntos en los ángulos obtenidos y tiempo transcurrido hasta ese momento, Así este puede hacerse una idea de las posturas que realiza y además pueda seleccionar uno de los puntos de la gráfica y le lleve justo al momento del video donde la realizó.

También podemos encontrar en la parte superior del menú un recuadro, que hace referencia al logo de nuestra aplicación.

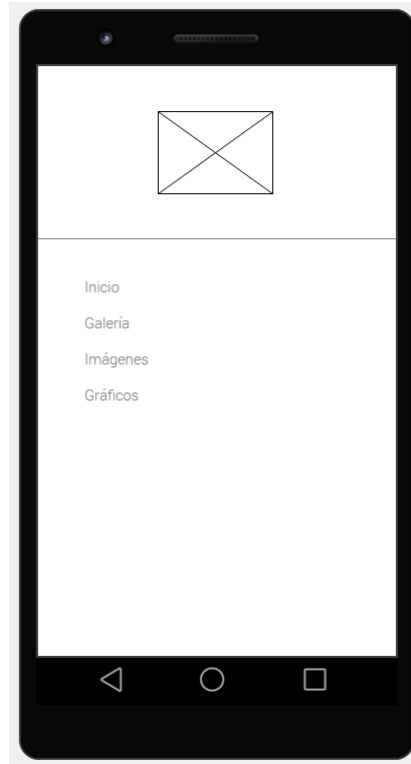


Figura 7.2: Mockup 2: Menú de la Aplicación

Una vez que ya habíamos desarrollado el menú y sabíamos cuales eran las secciones principales que teníamos que desarrollar comenzamos creando las vistas de los vídeos realizados “*Galería*”, los frames obtenidos de los vídeos “*Imágenes*” y los gráficos resultantes del posterior análisis logrado “*Gráficos*”. Para ello se nos ocurrió que una manera fácil de presentar el contenido existente de cada sección era a modo de lista, donde fueran apareciendo uno debajo de otro cada contenido realizado con el fin de poder seleccionar el que nos interesara.

Como se puede observar, las tres vistas se diseñaron exactamente igual 7.3, con la misma estética y con el mismo planteamiento de funcionalidad. Lo único que iba a diferenciarse de una a otra era el contenido que se presentaba: en la primera mostrando los videos realizados, en la segunda los frames obtenidos de los vídeos y en la tercera después de haber realizado anteriormente el análisis se podría visualizar en ella un nuevo gráfico.

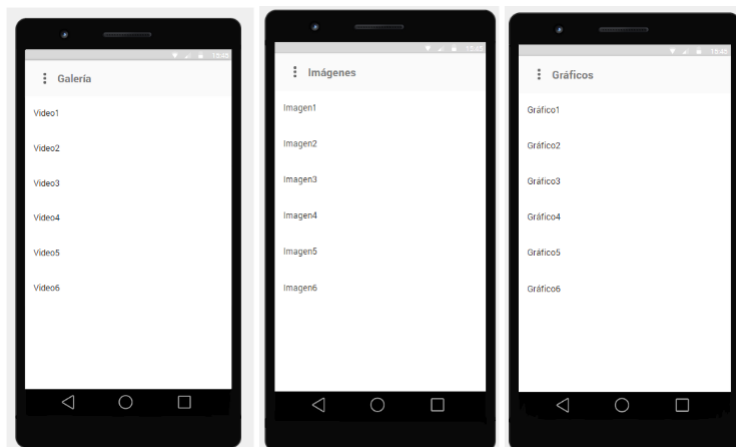


Figura 7.3: Mockup 3: Listas de la Galería, Imágenes y Graficos de la Aplicación

Ya creadas las tres vistas anteriores 7.3, ahora lo que se nos planteó era como diseñar la vista que aparecería después de la selección de cada contenido.

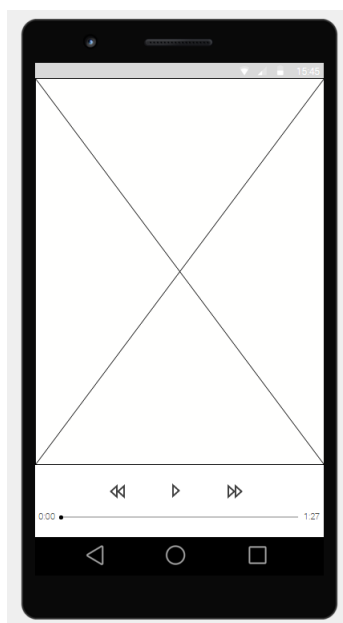


Figura 7.4: Mockup 4: Vista de la reproducción de los vídeos realizados

Así que siguiendo el orden del menú 7.2, comenzamos con la vista de la “Galería”, donde se encuentran todos vídeos que hemos realizado. Como nuestro objetivo era lograr que una vez seleccionado cualquiera de los vídeos presentes en el listado lo que tenía que suceder era que se reprodujeran, decidimos crear una vista previa de un reproductor, tal y como se muestra en la figura, 7.4 con las opciones de reproducción, retroceder, avanzar y pudiendo observar en qué momento del video nos encontrábamos.

Todo lo diseñado en la aplicación, ha sido siempre analizando y teniendo en cuenta cuáles podrían ser las necesidades y aspectos fundamentales de nuestros usuarios. Para que obtengan una aplicación completa, útil y que se adecue a sus requisitos y objetivos.

Una vez terminada la vista anterior, seguimos con la vista de “*Imágenes*” 7.5, en la que se encuentran todos los frames obtenidos de los vídeos seleccionados. Por lo que teniendo ya hecho la vista del listado de los frames, ahora de lo que nos teníamos que encargar era de diseñar la nueva vista a la que se accediera seleccionando uno de los frames del listado. Así que creamos una vista, aún sabiendo desde un principio que seguramente no fuera la definitiva debido a posibles problemas, cambios de decisiones o circunstancias que se nos pudieran dar a la hora de analizar el video, pero teníamos que partir de algún modo de un diseño base de como pensábamos que íbamos a filtrar los leds de la imagen, con lo que la generación de este prototipo tenía un papel fundamental para nuestra aplicación.

Al final, después de pensar cómo íbamos a analizar el vídeo y a detectar en una imagen los leds incorporados por el ciclista sabíamos que una actividad principal de la aplicación sería filtrar la imagen por RGB con el color de los leds para poder aislarlos. De manera que, como se puede observar en la figura 7.5, creamos provisionalmente tres sliders para ofrecerle dicho filtrado al usuario. Además colocamos un botón en la esquina inferior izquierda de está vista, con la idea de que una vez que se haya procedido a la filtración, la persona pudiera aceptar dichos parámetros y proceder al análisis del video correspondiente.

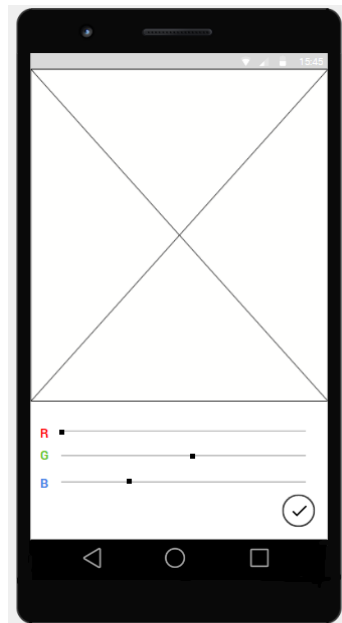


Figura 7.5: Mockup 5: Vista de los frames obtenidos y filtrado de los leds

Y el último mockup que realizamos fue la vista donde se iba a representar el gráfico

de nuestro análisis realizado.

Pensamos, que principalmente una de las cosas que tendríamos que incorporar en ella era en el eje “X” la progresión del tiempo y sobre el eje “Y” la de los ángulos, para que así el usuario se pudiera hacer una idea de en qué momento ha ido realizando dichas posturas.

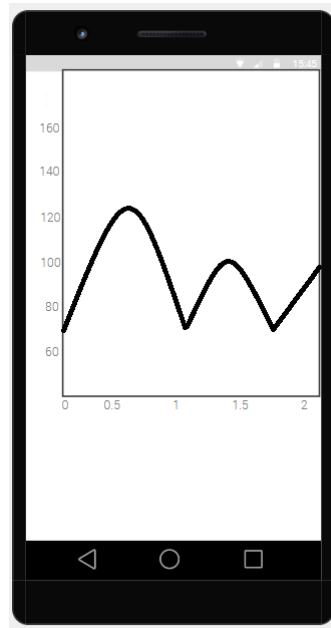


Figura 7.6: Mockup 6: Vista del gráfico resultante

7.2. Aplicación actual

La intención de este apartado, es mostrar cómo ha quedado finalmente el diseño de nuestra aplicación y explicar razonadamente el porqué de los cambios en el diseño final frente al prototipado mostrado en la sección anterior.

A la hora de presentar los diseños finales, vamos seguir el mismo orden que en el apartado anterior, puesto que a la hora de implementar y desarrollar la interfaz gráfica de nuestra aplicación, fue el orden que llevamos a cabo, creando primeramente las secciones de nuestra aplicación más relevantes e imprescindibles.

Así que siguiendo con lo comentado, el primer diseño final que se va a presentar es

la vista previa de la cámara, esta sección para nosotros ha terminado llamándose en nuestra aplicación con el nombre de *Inicio*, puesto que como hemos comentado anteriormente, era una sección imprescindible y el punto de partida de nuestra aplicación a la hora de comenzar a realizar el análisis biomecánico a un ciclista.

En variación con el diseño del mockup que realizamos previamente 7.1 para esta vista, podemos encontrar ligeras diferencias, como la utilización finalmente de un solo botón, con el objetivo de ocupar el menor espacio posible de pantalla y hacer más fácil el uso de la cámara al usuario, ya que pensábamos que quedaría más claro en este caso cuál es el botón de “*rec*” y cuál el de “*stop*” y en qué momento se está grabando, ya que van apareciendo alternativamente, debido a la selección de cada uno de estos.

Para mayor aprovechamiento de pantalla, aparte de solo dejar un botón también procedimos a dejar transparente el toolbar superior e inferior de la pantalla.

Por último, otro de los cambios llevados a cabo es la aparición de un *cronómetro* mientras se está grabando, para que así se pueda diferenciar en todo momento no solo por los botones que se muestran sino también por la aparición de este, si se está grabando o no, y así poder conocer de cuanto está siendo la grabación del vídeo mientras se está llevando a cabo.

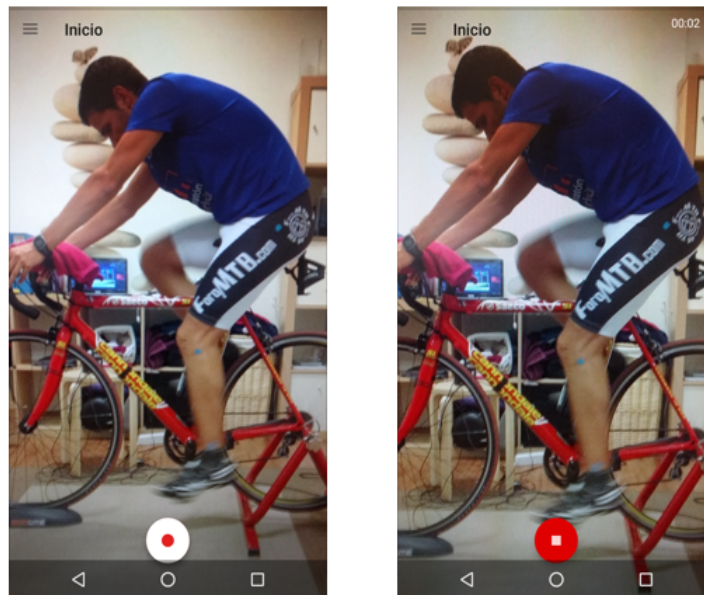


Figura 7.7: Pantalla de Inicio - Realización de la grabación del vídeo

La realización de la anterior vista expuesta, *Inicio*, nos llevó a pensar la necesidad que teníamos de dotar de permisos a nuestra aplicación, circunstancia que no habíamos tenido en cuenta cuando comenzamos a desarrollar el prototipado.

Por ello terminamos creando la siguiente vista 7.8, la cual aparece al comienzo de nuestra aplicación en el caso de no tener activados los permisos que se necesitan, para avisar al usuario que estos son imprescindibles.

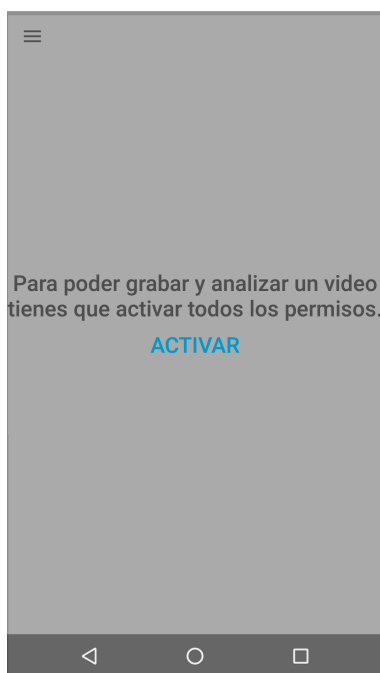


Figura 7.8: Activar Permisos

Los tres permisos necesarios en nuestra aplicación son:

- Cámara
- Almacenamiento
- Micrófono

Los cuales se representan mostrados en grupo, y desaparecerán una vez que hayan sido aceptados, mostrándose solo entonces lo que están pendientes de activación.

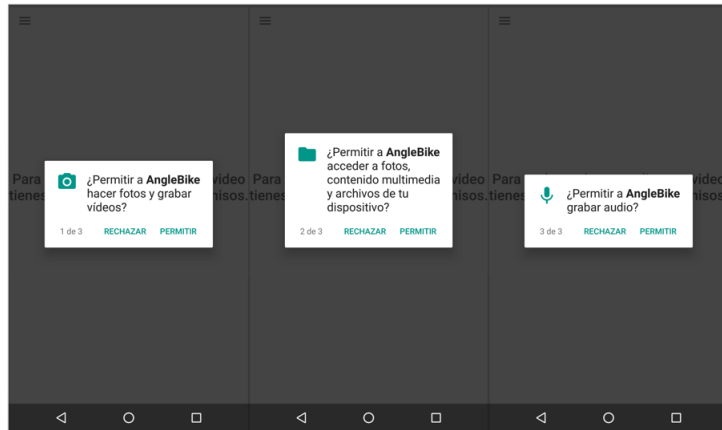


Figura 7.9: Permisos: Cámara, Almacenamiento y Micrófono

La tercera vista que vamos a mostrar es la del *menú*, al final una vez que nos pusimos a desarrollar esta interfaz, nos dimos cuenta que podíamos incluir en vez de un menú diseñado con los tres puntos como anteriormente se ha mostrado en los mockups 7.2, el del diseño de las tres rayas, el cual lo incluimos programando, siguiendo las pautas de diversas páginas que nos proporcionaban aprender a cómo hacerlo [DRA].

Como se puede ver, finalmente nuestro menú incluye más secciones que el que previamente diseñamos en nuestros prototipos 7.2. A continuación, vamos a comentar que es lo que presenta cada apartado.

- Inicio: Explicado anteriormente 7.1, es el punto de partida para comenzar a realizar nuestro análisis biomecánico, en él se procede con la grabación del ciclista.
- Analizar: Donde podemos encontrar un listado de frames obtenidos de cada uno de los vídeos grabados anteriormente, es decir, que por cada vídeo grabado y guardado en la sección *Galería* aquí podremos encontrar un frame correspondiente a cada uno de ellos para finalmente poder seleccionar uno de los frames presentes en la lista, proceder al recorte de la sección de la imagen que nos interesa, la cual será la parte donde se encuentren situados los leds en el ciclista y después de haberla seleccionado pasar por último a filtrar la imagen resultante para que solo se queden visibles los leds y ya así poder comenzar con el análisis.

- Galería: Donde podríamos encontrar un listado con todos los vídeos que hemos realizado de nuestras actividades para posteriormente seleccionar uno de ellos y poderlo reproducir.
- Gráficos: En esta sección podremos encontrar un listado con los archivos que se han analizado, y en el caso de seleccionar uno de ellos, se nos mostrará un gráfico en el cual en el eje “X” se representa el tiempo trascurrido en el vídeo y en el eje “Y” la progresión de los ángulos alcanzados por el usuario, y una gráfica de puntos resultantes, con el ángulo que se ha obtenido en cada momento, con el fin de que nuestro usuario si selecciona alguno de los puntos de la gráfica, se le reproduzca inmediatamente el momento del vídeo en el que sucedió y se adoptó dicho ángulo.
- Resultados: Donde podríamos encontrar un listado con todos los vídeos ya analizados, con el fin de seleccionar uno de ellos y poderlo reproducir y ver los ángulos que se van adoptando en cada momento.

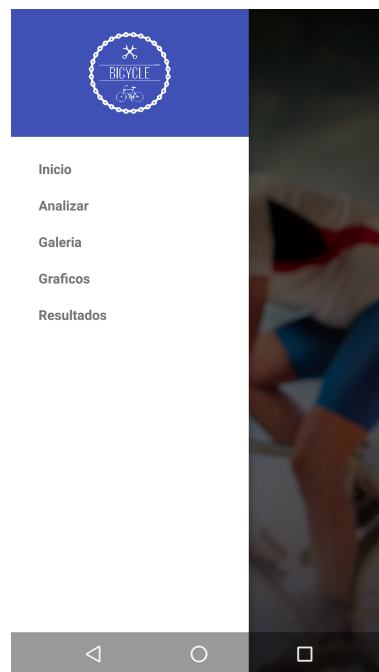


Figura 7.10: Menú

Una vez realizado el menú procedimos a realizar la vista de la *Galería*, debido a que todos los vídeos realizados por la cámara teníamos que irlos almacenando y mostrando en algún sitio, por ello fue la siguiente sección que comenzamos a realizar.

En un principio, se diseñó a modo de lista 7.3, es decir, mostrándose todo el contenido visualmente uno debajo del otro porque pensábamos que era una manera fácil y sencilla de representar la información visualmente, pero finalmente cambiamos de opinión y terminamos mostrándolo en cuadrículas, lo que se conoce en Android como *Grid View* [GRI]. Porque nos guiamos por la apariencia visual de la galería interna de Android.

Una vez realizada la Galería, decidimos que el resto de las secciones *Analizar*, *Gráficos* y *Resultado*] que precisaban de mostrar también un listado de contenido, se iba a mostrar finalmente de la misma forma que la Galería actual, todo cuadrículado y con diferentes imágenes, según la temática del contenido de la sección.



Figura 7.11: Grid View, Galería, Analizar, Gráficos y Resultados

Ya realizada la *Galería*, el siguiente paso era la reproducción de los vídeos que habían sido grabados por la cámara de la aplicación. Habiendo dado ya la opción al usuario a que pudiera seleccionar cualquiera de los vídeos que se mostraban en la cuadrícula, quedaba por realizar la vista del reproductor de vídeo.

El reproductor, se realizó totalmente igual que el plasmado en un principio en los prototipos anteriormente expuestos 7.4. Este contaba con la opción de reproducir, retroceder, adelantar y el slider que va avanzando según el tiempo que va transcurriendo en la reproducción.



Figura 7.12: Reproductor de los vídeos realizados

Por consiguiente, comenzamos con la realización del apartado de *Analizar*, el cual sufrió muchas modificaciones a lo largo del desarrollo de la aplicación y además es lícito resaltar que el diseño que obtuvimos y aquí se presenta tiene un peso fundamental en la optimización del análisis de vídeo.

Finalmente, para que nuestro usuario pueda proceder a realizar un análisis de los vídeos que anteriormente ha grabado tendrá que seleccionar uno de ellos de la vista de *Analizar* ya presentada 7.11 y a partir de ahí, tendrá que realizar dos pasos más antes del análisis.

El primero de ellos es el que podemos ver a continuación 7.13, en esta vista se presenta un frame obtenido del vídeo que el usuario eligió, con el fin de recortar la sección del frame que realmente nos interesa, que es la zona donde se encuentran situados los leds del ciclista. Esto que aquí realizamos tiene una gran importancia y relevancia en nuestro proyecto, porque conseguimos que en vez de procesar el vídeo con sus dimensiones reales, se procese solo la zona que verdaderamente nos interesa, agilizando así la espera del usuario.

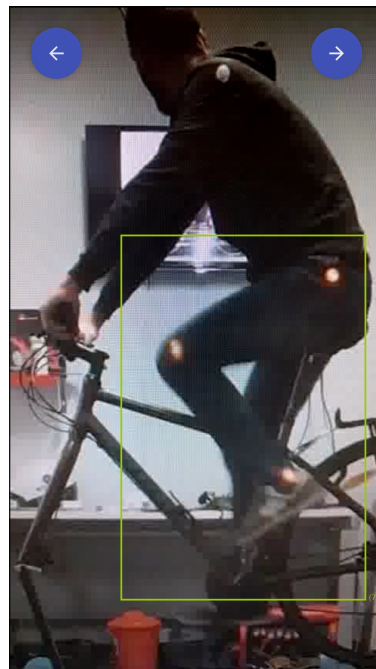


Figura 7.13: Analizar: Recorte de la sección necesaria

El segundo paso sería proceder a resaltar solamente los leds en la imagen, oscureciendo el resto de los detalles de esta. Por ello procedimos a realizar el segundo diseño tal y como se muestra en la figura 7.14, mostrando primeramente la imagen resultante del recorte de la pantalla anterior, y luego los sliders que son los que le permiten al usuario ir modificando en tiempo real la imagen para dejar mostrando solamente los leds. Estos sliders se pueden ir modificando mediante el arrastre del dedo por la línea que conforme al slider, o también decidimos darle la comodidad al usuario de que pudiera insertar una cantidad de 0 – 255 en los márgenes derechos. Con ambos procedimientos llegamos a obtener el mismo resultado.

Los botones colocados los márgenes superiores, se hicieron de esa forma para aprovechar espacio, el que se muestra en la esquina superior izquierda nos facilitará la opción devolver a la pantalla anterior en el caso de que queramos recortar de nuevo el frame y el situado en la esquina superior derecha se creó con el fin de que el usuario pueda proceder al análisis, lo que conlleva a la aparición de un *pop-up* como se puede ver en la figura, que le mostrará al usuario el porcentaje de vídeo analizado.

Como se puede ver el diseño del mockup de esta sección 7.14, no tendría nada que ver con el anteriormente definido 7.5, pero hemos tenido que ir adaptando la eficiencia y la optimización del análisis progresivamente y realizando cambios en la interfaz gráfica cada vez que descubríamos que podíamos lograr una mayor eficiencia.

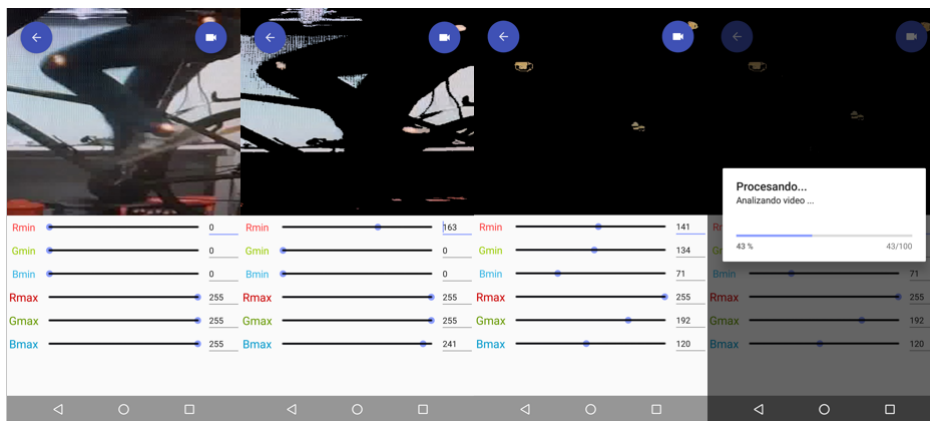


Figura 7.14: Analizar: Degradado de imagen

Lo siguiente que realizamos fueron los *Gráficos*, parte fundamental en la que se visualiza el análisis realizado.

Para esta parte lo que creamos fue primeramente la vista de cuadrícula anteriormente expuesta 7.11, en la cual la selección de uno de los componentes nos transportaría a las vistas que ofrecemos a continuación.

Nuestro propósito como reflejamos en los mockups 7.6, lo logramos conseguir tal y como se puede ver, utilizando la librería *GraphView* [GRAB], la que nos proporcionó la opción de realizar los gráficos resultantes. Y además para vídeos de una larga duración, en los que el análisis es largo y exhaustivo y cuya presentación en la imagen es muy conglomerada, logramos que con el gesto de pinza de los dedos

sobre el gráfico este se puede expandir y deslizar para ver todo con más claridad.

También logramos que la selección de uno de los puntos rojos que se exponen en la gráfica nos transporte a la reproducción del vídeo resultante en el momento exacto en el que se realizó dicho ángulo, con el fin de dotarle al usuario de una mayor capacidad información y de control de sus movimientos.

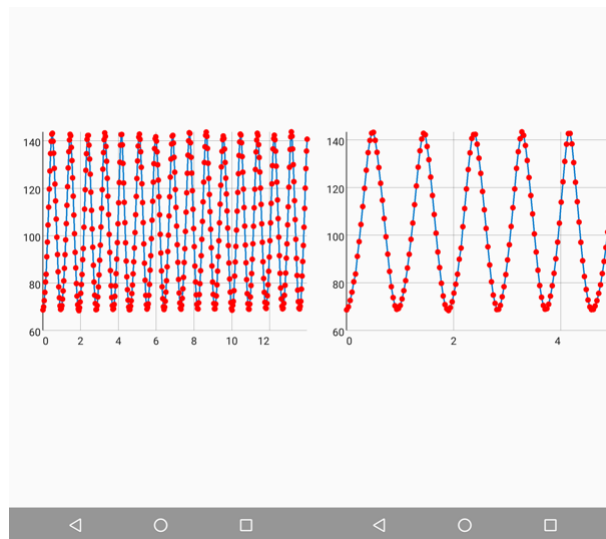


Figura 7.15: Gráficos

Para finalizar, la última vista que os presentamos que se nos ocurrió realizar con el fin de mejorar nuestra aplicación y dotarle de más información y datos al usuario es *Resultados*.

Cómo ha ido sucediendo a lo largo de nuestra explicación, en esta interfaz también se elegiría uno de los vídeos que se muestran en la cuadrícula, y después se pasaría a reproducirse.

La interfaz gráfica de esta sección es igual que la del reproductor ya utilizado 7.12, lo único que aquí se muestran los vídeos procesados, en los que se dibujan en cada momento los ángulos que se van formando en la realización de los movimientos.

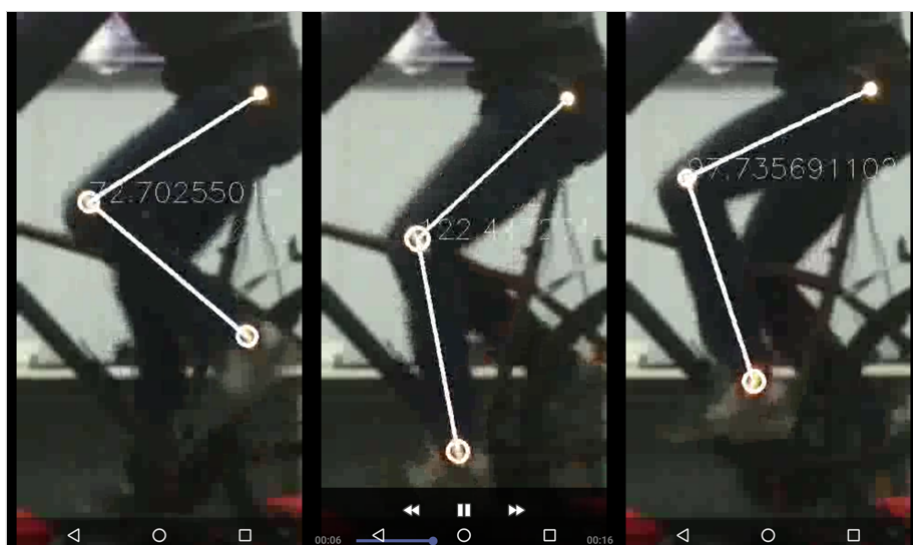


Figura 7.16: Reproducción de los vídeos resultantes del análisis

Capítulo 8

Dificultades encontradas y sus soluciones

A continuación os vamos a comentar los problemas con los que nos hemos encontrado a la hora de ir desarrollando nuestra aplicación y las soluciones que hemos optado a tomar, estos se van a ir mostrando en orden cronológico de creación de nuestra aplicación, es decir, empezando desde el primer problema que encontramos cuando comenzamos con la realización del trabajo hasta su posterior finalización.

8.1. MediaRecorder: start failed: -19

Uno de los primeros pasos fundamentales que tuvimos que dar cuando comenzamos con la creación de la aplicación Android era hacer una sección que nos permitiera grabar en vídeo, para ello optamos por utilizar la cámara nativa de Android con sus clases más conocidas; *Camera* [CAMA] que nos permite abrir y manipular los parámetros de la cámara del dispositivo, y su clase *MediaRecorder* [MEDb] fundamental para poder grabar y que nos diera además cierto control sobre la calidad de vídeo y audio pudiendo modificarlos a nuestro gusto.

El problema lo encontramos en la comunicación de estas dos clases, es decir, cuando por un lado era imprescindible la utilización de *Camera* para la previsualización de la imagen, como cualquier cámara de un dispositivo móvil cuando se inicia y te muestra la imagen sin tener que proceder a la grabación simplemente se nos

muestra lo que está captando el objetivo de la cámara, pero cuando procedemos a nuestra grabación de vídeo pulsando el botón de grabar que se encuentra en la pantalla, ahí entra en conflicto la clase Camera con la clase MediaRecorder qué es la que tiene que comenzar con la grabación.

Lo que nos sucedía es que cuando nuestra aplicación comenzaba a funcionar y se previsualizaba lo que captaba la cámara, si nosotros procedíamos a presionar el botón de grabación la ejecución dejaba de funcionar repentinamente y el único error informativo que se nos mostraba en el programa Android Studio a la hora de la depuración era: *"start failed: -19"*. Estuvimos investigando durante varias semanas y nos encontramos con que muchas más personas, se había encontrado con errores semejantes, como *"prepare failed: -1"* [ERRa] , *"start failed: -38"* [ERRc] y *"start failed: -16"* [ERRb], estos no se encuentran documentados en ningún sitio, es decir, no existe ninguna página en la que se haya redactado el significado de cada uno de los distintos fallos, pero sin duda el error más común que nos encontramos era *19*".

Investigando durante esas semanas, nos encontramos también con un desarrollador que había publicado en una entrada de su blog [BLO], en la que afirmaba que el problema residía en que MediaRecorder internamente cambiaba las dimensiones de la pantalla haciendo que no coincidieran con las dimensiones que previamente habíamos fijado nosotros a la clase Camera. Por lo que decidimos probarlo pero esto no solucionó el problema.

Decidimos examinar más a fondo la página de Android Developers referido a Camera API [API], en la cual pudimos ver que para utilizar la cámara nativa de Android correctamente teníamos que llamar a los métodos en un orden estricto, lo cual no cumplíamos del todo, ya que nos faltaba una llamada al método Camera.unlock(), cuya función es la de desbloquear la cámara para que otros procesos puedan usarla, en nuestro caso el error se producía porque mientras que la cámara estaba siendo usada por Camera, MediaRecorder.start() intentaba también usarla sin haber sido desbloqueada previamente.

8.2. Cabeceras nativas C++ (JNI Android Studio)

Otro de los principales errores que se nos presentó fue cuando tuvimos que importar la librería de OpenCV correctamente, para poder utilizar su código en nuestro

proyecto de Android Studio, lo primero que hicimos fue buscar una guía de instalación en su página oficial [OPE], pero descubrimos que estaba desactualizada y solo lo tenían para proyectos Android en la plataforma Eclipse, por lo que tuvimos que buscar información en otras paginas y foros. Al final encontramos una web [PROa] en la que nos explicaba cómo importar el proyecto, el resultado de esto es que se podía llamar a las clases y hacer código sin problema, pero al compilarlo generaban múltiples errores relacionados con el código nativo.

Después de investigar entendimos que OpenCV internamente hacía llamadas a cabeceras C++, este es el lenguaje original en el que fue escrita la librería, ocasionaba errores en tiempo de compilación.

Para poder solucionarlo, el módulo SDK de OpenCV aparte del proyecto que se importa también te proporciona las librerías8.1 JNI para las diferentes CPUs para las que han salido los dispositivos Android, cada una de estas tiene una interfaz distinta por lo que se necesitaba copiarlas todas en la carpeta JNILibs de Android Studio para se pueda compilar en cualquier dispositivo.

```
OpenCV-android-sdk
├── doc
├── samples
├── sdk
│   ├── etc
│   ├── java
│   ├── native
│   │   ├── 3rdparty
│   │   ├── jni
│   │   ├── libs
│   │   ├── armeabi
│   │   ├── armeabi-v7a
│   │   └── x86
│   └──
├── LICENSE
└── README.android
```

Figura 8.1: Ejemplo de la estructura de paquetes de OpenCV4Android

8.3. Motivo y razón del uso de la orientación vertical de nuestra aplicación

A continuación vamos a exponer la causa de esta determinación, primeramente decir que el hecho de decidir fijar una orientación no es debido a un error encontrado, ni a un problema, ni impedimento a la hora de realizar nuestro trabajo, por ello queremos que conste una explicación razonada del porqué tomamos dicha decisión.

En un principio, la aplicación se pensó que dotaría de ambos modos de rotación, para darle siempre a elegir al usuario la posibilidad de su utilización y grabación de los videos de ambas maneras. Pero una vez que íbamos avanzando en su desarrollo, nos dimos cuenta que si permitimos al usuario realizar la grabación en modo horizontal, el video iba a captar muchos objetos, formas y brillos innecesarios, que podrían hacer que nuestro análisis posterior de los frames del video fueran mucho más impreciso.

Asimismo, reflexionamos si podríamos entorpecer al usuario o dificultarle a la hora de realizar la grabación captando la figura del ciclista, pero nos dimos cuenta de que tampoco, ya que lo indispensable es poder percibir y obtener las extremidades inferiores del ciclista, donde se encuentran situados los leds que tenemos que detectar.

Así que después de analizar todas las consecuencias que dicha decisión tendría en nuestro trabajo, pudimos ver que era totalmente viable y favorable fijar la orientación vertical, ya que con ello conseguiríamos ofrecer al usuario una mayor precisión y fiabilidad.

8.4. Problema de lectura del video por OpenCV

Aun teniendo todas las funciones de OpenCV disponibles nos encontramos con un problema, el lector de videos de OpenCV en Android solo es capaz de leer un formato determinado, que no es el que crea la cámara del propio Android (respuesta que hallamos en el foro de respuestas de OpenCV) [LEC]. Debido a esto tuvimos que recurrir a JavaCV, que poseía herramientas que nos permitían la lectura del video correctamente.

En primera instancia usamos JavaCV para leer el video grabado por la cámara y convertirlo al formato adecuado para OpenCV. Esto sin embargo suponía una espera innecesaria al usuario, ya que obligaba a éste a esperar después de grabar el vídeo a que se convirtiera al formato adecuado mientras la aplicación quedaba bloqueada hasta finalizar el proceso.

Después de varias pruebas llegamos a la conclusión de que el tiempo de espera solo podíamos introducirlo en el paso final, el análisis del video, así que en este paso realizamos las conversiones necesarias a la vez que realizamos el análisis.

8.5. Importación de OpenCV y JavaCV en Android Studio

Debido a la necesidad de utilizar funciones de OpenCV y de JavaCV juntas, tuvimos que importar las dos librerías en Android Studio, pero al momento de ejecutar el código, este devolvía un error parecido al que teníamos cuando no incluimos las librerías relacionadas con la arquitectura del CPU.

El problema era que configuramos el Gradle Build [GRAa] de la aplicación para que se incluyera JavaCV de forma automática descargandose de internet, esto importaba tanto los fichero con las clases como todos los fichero internos de la arquitectura del procesador que en este caso se encontraba en el javacpp. Por lo que posteriormente al importar manualmente el módulo de OpenCV, el entorno pasaba a ejecutar las librerías internas de este para los dos módulos, esto resultaba error ya que aunque tuviesen el mismo nombre por estar dirigidas a las mismas arquitecturas CPU de los procesadores móviles, internamente mientras unas llevaban instrucciones Bydecode para la máquina virtual de java, la otra necesitaba las cabeceras JNI para poder interactuar con programas escritos en código nativo, por lo cual ninguna de las dos eran compatibles. La solución se encontró en la página GitHub de JavaCV [IMPa], y consistió en importarlo de forma manual copiando los ficheros de la arquitectura en las carpetas correspondientes del proyecto.

8.6. Problemas de rendimiento

Al realizar una aplicación de análisis visual en un dispositivo móvil no se dispone de la misma potencia que un ordenador por lo que el tiempo de ejecución se ve considerablemente aumentado. La aplicación en un principio no era capaz de estar procesando más de un frame por segundo, y si el video era demasiado largo se cerraba automáticamente ya que la memoria se desbordaba.

Esto lo fuimos resolviendo poco a poco mediante la optimización del código de análisis, primero controlamos mejor las creaciones de objetos en memoria y re-utilizamos algunos para varias tareas, especialmente las matrices usadas para las imágenes que eran las que más carga tenían, más tarde separamos el video en imágenes individuales en vez de analizar el video al completo. Con esto conseguimos llegar a casi 3 frames de análisis por segundo reduciendo bastante el tiempo de espera.

Capítulo 9

Manual de la aplicación

En este epígrafe se describe el funcionamiento que tendrá la aplicación, para facilitar la experiencia del usuario y que no haya confusión durante la ejecución de la misma, ya que no es una aplicación de cámara de grabación de vídeo convencional.

9.1. Grabación y reproducción

Para empezar, en la primera pantalla te encuentras con la cámara de vídeo preparada para grabar si se pulsa el botón con el círculo rojo en el interior, pero antes se aconseja al usuario que coloque el dispositivo móvil en posición vertical, esto se ha pensado así para que se pueda grabar lo más cerca posible de la persona montada en la bicicleta estática de manera que se pueda recoger una buena iluminación de los leds y una buena toma en el que pueda caber todo su cuerpo, facilitará mucho el análisis posterior que se va a realizar. Después de empezar a grabar, aparecerá en la parte superior derecha de la pantalla la duración que lleva el video. Cuando el usuario considere que debe parar la grabación entonces se presionará una sola vez el botón rojo, que permanecerá de este color hasta que termine de hacer los procesos internos de conversión, esto se sabrá cuando vuelva otra vez al botón blanco que tenía inicialmente. La secuencia de esta ejecución se puede ver en la figura 9.1.

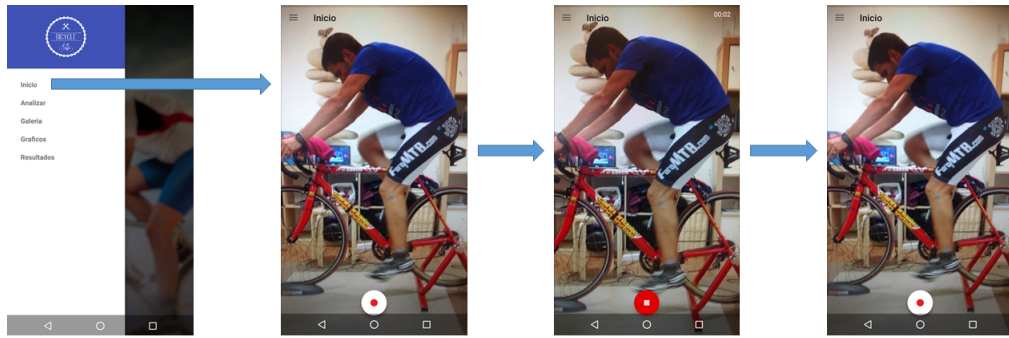


Figura 9.1: Sección Inicio - Grabación

Después de generar el vídeo se accede al menú lateral izquierdo de la pantalla para cambiar a la vista de *Galería* y poder reproducirlo, como se observa en la figura 9.2, y así poder comprobar que el resultado es analizable. Se considera que un resultado es analizable si a lo largo de todo el video se puede observar perfectamente el movimiento de los leds, si no es así, se aconseja hacer una nueva prueba volviendo a la pantalla de *Inicio* de la cámara.

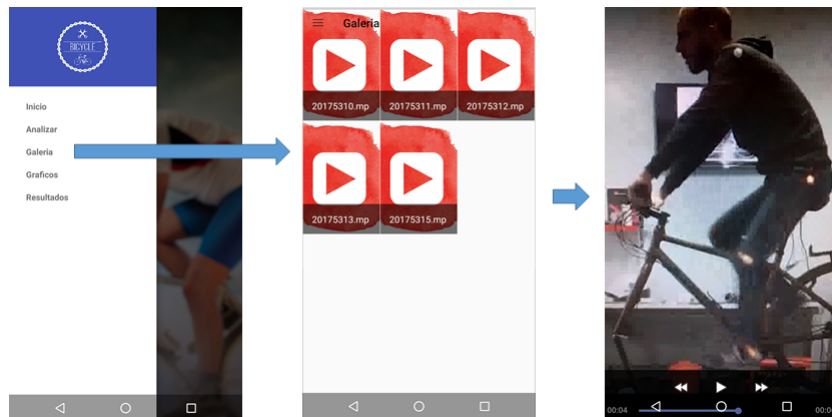


Figura 9.2: Sección Inicio - Reproducción

9.2. Análisis del vídeo

En esta parte del proceso se va a mostrar la secuencia de cómo realizar el recorte y el degradado de la imagen, para luego en función de estos valores se obtenga el

nuevo video modificado con los ángulos.

9.2.1. Recorte del vídeo

En el caso de que el vídeo grabado en la sección anterior se considere se considere que es analizable, entonces se cambiará a la vista de *Analizar* y se va buscará la imagen con el mismo nombre del vídeo resultante, una vez seleccionado se pulsará el botón siguiente situado en la parte superior derecha que lleva a la siguiente vista del análisis en el que se hará el recorte de la imagen. El objetivo es reducir el tamaño de la imagen y eliminar un marco exterior que no es relevante y que además hace más lento el procesado posterior, la idea es dibujar una figura geométrica en la imagen que va ayudar a ver el recorte que se va a producir, para ello se va situar el dedo en la parte superior izquierda de la pantalla lo máximo que se considere pero sin llegar a tocar su vértice y asegurándose de no alcanzar ninguno de los leds, una vez se ha elegido la posición, sin levantar el dedo se va a desplazar hacia abajo en dirección diagonal a la esquina inferior derecha pero sin llegar del todo para finalmente levantar el dedo, como solución final se debe ver un rectángulo de color amarillo que delimita la sección que se va obtener y en cuyo interior deben estar todos los leds completamente enteros. Si se cumplen estas condiciones entonces se pulsa el botón superior derecho para confirmarlo. Esta parte de la pantalla de Análisis se puede ver en las tres primeras imágenes de la figura 9.3.

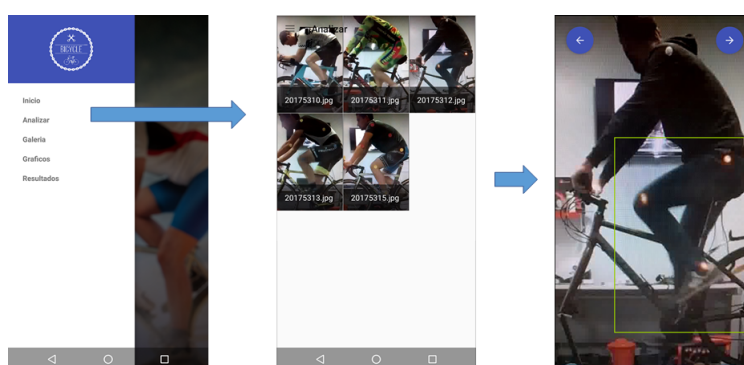


Figura 9.3: Sección Analizar - Recorte del vídeo

9.2.2. Degradado de la imagen y generación del vídeo

La vista que aparece a continuación divide la pantalla en dos partes, en la parte superior se ve la imagen con el recorte previamente hecho y en la parte inferior los sliders con los que se podrá cambiar los valores máximos y mínimos de los píxeles de la imagen, aunque también se tiene opción de hacerlo con los campos de texto situados al final de cada slider para ser más precisos.

La idea de esta pantalla es degradar los colores de la imagen jugando con los slider hasta obtener de objetivo una nueva imagen en la que se solo sean visibles los leds con un fondo totalmente oscuro o lo más oscuro que se pueda llegar, para este desarrollo se tiene la imagen recortada de la parte superior que es usada como vista previa en tiempo real de lo que sucede mientras se van cambiando los valores de los sliders o de los campos textField.

Para finalizar se presiona el botón circular de la parte superior derecha con la miniatura de una cámara, esto indica que se van a calcular los ángulos del vídeo , mientras sucede el proceso se ve una ventana de carga superpuesta en el que se va a informar al usuario del porcentaje de vídeo que lleva analizado, cuando llegue al 100 % significa que ha terminado de reconstruir el nuevo vídeo con los ángulos. Se puede observar mejor en las dos últimas imágenes de la interfaz 9.4.

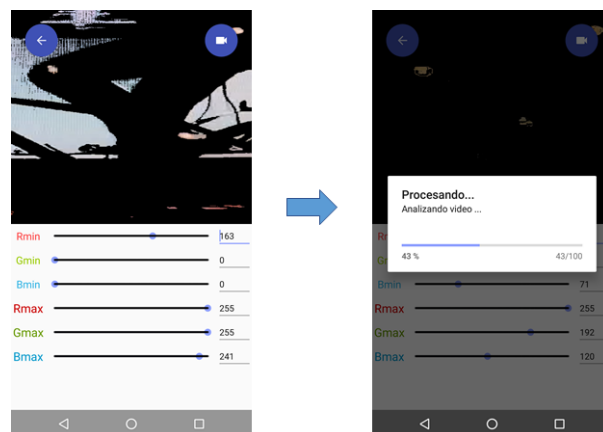


Figura 9.4: Sección Analizar - Degradado de la imagen y generación del vídeo

9.3. Resultados

Al terminar el proceso la siguiente pantalla que aparece es la cámara, o la de *Inicio*. Llegados a este punto se pueden tomar dos caminos a decisión del usuario, se puede ver el gráfico generado o el vídeo reconstruido ambos con los ángulos. Para acceder a cualquiera de las opciones se despliega el menú izquierdo y se elige cualquiera de las dos.

Suponiendo que se elija la ventana de los gráficos, esta consiste en una sencilla interfaz en la que se muestran todos ordenados en cuadrícula. Se accede al gráfico que se desea buscando por el mismo nombre del vídeo que se realizó, que además para no confundir al usuario lleva la fecha del día en el que se generó. Cuando se encuentra el gráfico se pasa a su interfaz propia, en ella se puede ver un plano en el que el eje horizontal representa el tiempo y el vertical el ángulo que se ha alcanzado en dicha duración, se puede hacer zoom con gesto de pinza para que los puntos sean más accesibles al usuario como se puede observar en la figura 9.5. Con los puntos rojos se puede interaccionar ya que si se pulsa se abre el vídeo pausado en el momento justo en el que se alcanzó dicho ángulo.

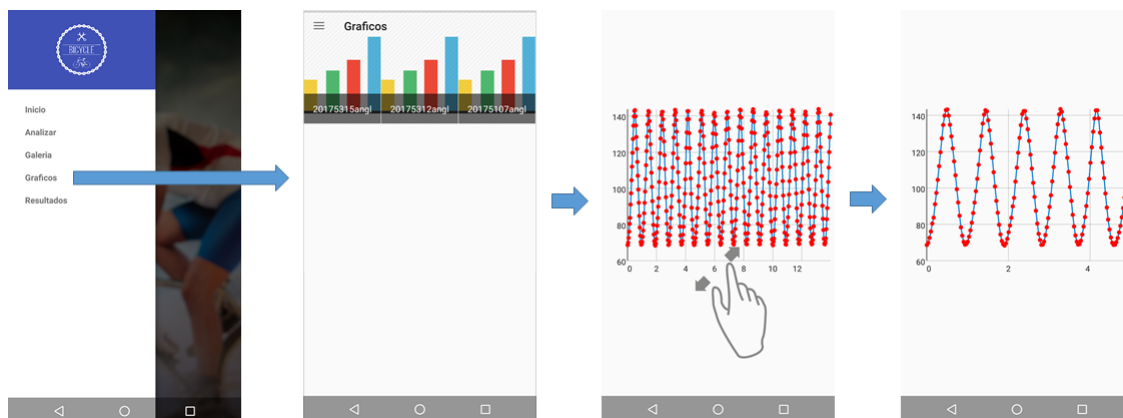


Figura 9.5: Sección Gráficos - Gráficos resultantes del análisis

Por otro lado, al vídeo modificado con los ángulos se accede mediante la opción de *Resultados* de la pantalla lateral izquierda. Los vídeos en esta interfaz se muestran ordenados de manera cuadriculada, en los que si se selecciona uno se abre un reproductor que muestra el vídeo grabado pero con los ángulos mostrados continuamente 9.6.

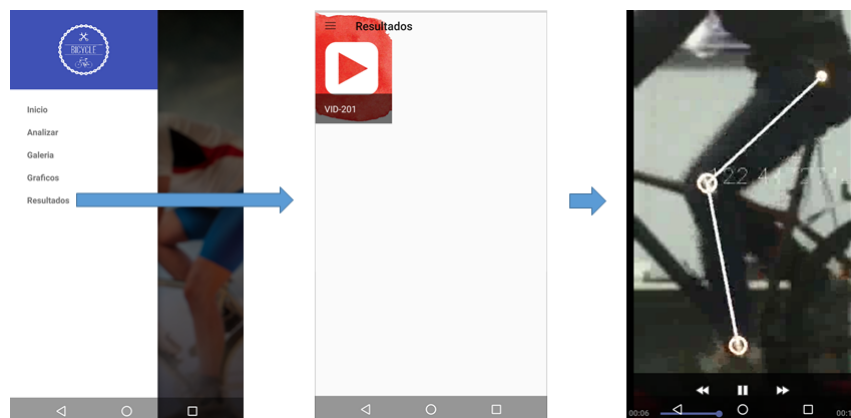


Figura 9.6: Sección Resultados - Vídeo resultante del análisis

9.4. Posibles problemas

En esta sección se va a explicar cómo se han controlado posibles problemas que pueden surgirle al usuario, estos problemas pueden ser que esté faltó de memoria y la falta de permisos necesarios de la aplicación. Se va explicar cómo se debe interactuar para solucionarlo o la secuencia que se debe seguir.

9.4.1. Memoria insuficiente

Si el dispositivo no dispone de memoria suficiente para que la aplicación pueda realizar un vídeo y además de no soportar los archivos que se generan después del análisis, entonces la aplicación muestra una ventana emergente con dos opciones para el usuario, con el botón de la izquierda se cierra la ventana y con el de la derecha se accede directamente al almacenamiento interno del dispositivo con el objetivo de ver el espacio que hay disponible. En la imagen 9.7 se puede observar la secuencia si se elige el botón derecha.

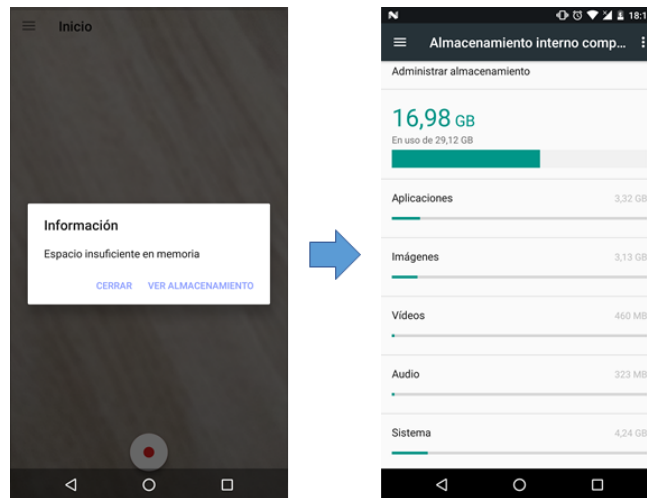


Figura 9.7: Memoria insuficiente

Como se puede observar dicho botón te redirige directamente a los ajustes del almacenamiento interno del dispositivo móvil, para que puedas observar la cantidad de espacio libre que le queda al teléfono y puedas administrarlo.

9.4.2. Permisos

Este es otro de los problemas que pueden surgir desde que se instala la aplicación en el dispositivo y no se conceden los permisos, aunque solo si la versión del teléfono es Android 6.0 en adelante. Los permisos que pide son los de la cámara, el almacenamiento y el micrófono.

Cuando se inicie la aplicación lo primero que comprobará será si tiene los permisos activados, esto es importante ya que en el caso de que no los tenga entonces se cargará una pantalla como en la figura 9.8, en esta se aconseja que se habiliten y proporciona un botón que pide todos los permisos que no se tienen activados. Si se conceden solo los de cámara entonces se iniciará la ejecución normal de la aplicación cargándose la ventana de inicio con la vista previa y su botón de grabación, si se presiona y no se dispone de los permisos de micrófono y almacenamiento entonces solicita dichos permisos y no se inicia la grabación.

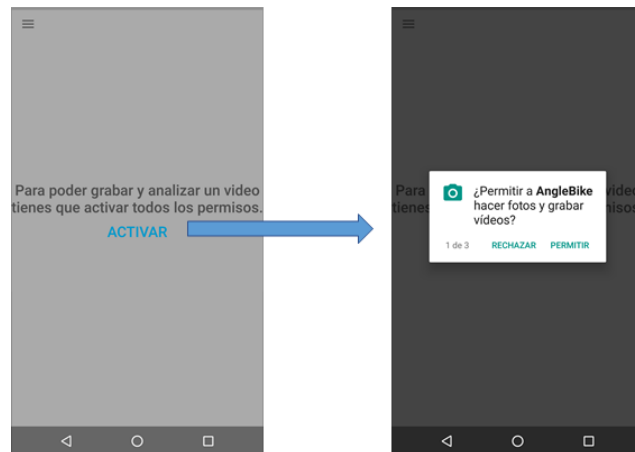


Figura 9.8: Habilitar Permisos

Si se decide seguir con la ejecución sin los permisos necesarios, el usuario podrá seguir interactuando con la aplicación pero con algunas limitaciones. En el caso de que se quiera acceder a algunas opciones del menú, se le pedirá el permiso que necesite la pantalla con una ventana emergente y un botón de acceso directo a los ajustes de la aplicación, para que pueda continuar la ejecución como se muestra en la secuencia de la figura 9.9.

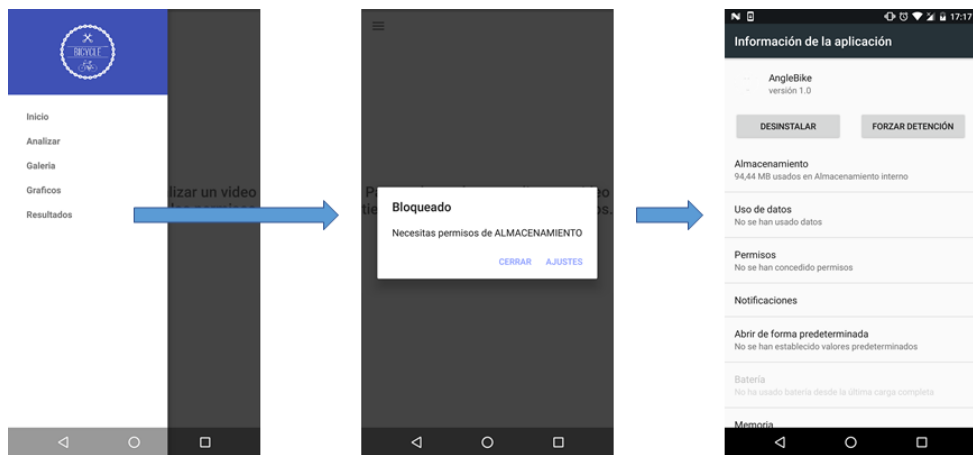


Figura 9.9: Redirección a ajustes para activación de permisos

Capítulo 10

Materiales y elementos necesarios para probar la aplicación

En este apartado vamos a explicar los materiales que son necesarios para poder probar AngleBike de forma correcta, ya que al ser una aplicación cuyo objetivo es conseguir un análisis del posicionamiento correcto del ciclista en una bicicleta es imprescindible contar con una serie de componentes y condiciones para que este se pueda llevar a cabo. Así que es necesario contar con:

- Leds que se colocarán en el lateral de la rodilla, el tobillo y en la cadera del ciclista tal como se ha podido ver en las imágenes de los apartados anteriores(7.7, 7.12, 7.13 y 7.14). Para qué así nuestra aplicación posteriormente pueda reconocerlos y realizar el análisis.
- La bicicleta sobre la que se quiere hacer el análisis, para poder adaptarla después a la posición correcta, para la evitación de lesiones.
- Rodillo [ROD], es decir un soporte donde colocar la bicicleta, tal y como se puede ver en la figura 10.1, para poder realizar el análisis en un sitio fijo, y que así se facilite tanto la grabación del vídeo, como el análisis.



Figura 10.1: Soporte para apoyar la bicicleta

- Un asistente que grabe al ciclista mientras realiza la prueba.

Para finalizar, queremos concluir con una recomendación para que nuestro análisis sea más eficiente y preciso, a la hora de realizar el vídeo es aconsejable proceder a hacerlo en un lugar en el que haya fondos oscuros, que no se divisen muchos objetos dentro de la imagen de grabación que contengan el mismo color que los leds que se están utilizando y no mucha cantidad de luz.

Capítulo 11

Funcionamiento interno de la aplicación

El funcionamiento de la aplicación se puede dividir en tres apartados claramente diferenciados los cuales se detallaran en los siguientes apartados.

11.1. Grabación del video

Primeramente nuestra aplicación utiliza la cámara integrada en el dispositivo para grabar un vídeo mientras el ciclista realiza sus movimientos.

11.2. Edición de imagen

Al usuario se le muestra una imagen que corresponde al primer frame del vídeo grabado. Aquí puede realizar un recorte del vídeo original para centrarse en los leds colocados y que el próximo paso le resulte más sencillo, este recorte se denomina la región de interés o ROI (Region Of Interest) [ROI].

El siguiente paso es utilizar los sliders de colores para filtrar la imagen o la ROI y aislar los leds, una vez aislados se procede a analizar el vídeo.

11.3. Análisis del video original y creación del video con los ángulos

En este paso se va a proceder a calcular los ángulos formados por el usuario durante el vídeo y a generar el vídeo resultado. Para ello se van a utilizar todos los datos aportados en los apartados anteriores 11.1 y 11.2 .

Esta parte es crucial y vamos a explicar detalladamente como funciona por dentro:

1. Se lee el video y frame a frame se va convirtiendo en imagen para ser leída por OpenCV.
2. Esta imagen se guarda en una matriz, que va a ser donde se reconstruya el vídeo. Si ha seleccionado una ROI se captura la submatriz correspondiente y si se analiza la imagen completa se crea una copia para ser tratada 11.1.



Figura 11.1: Imagen Original

3. Se aplican una difuminación de la imagen y operaciones morfológicas para mejorar la precisión 11.2.

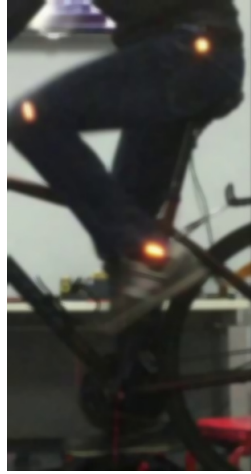


Figura 11.2: Imagen con Filtros

4. Con los datos de los sliders se calculan las partes de la imagen que están en el rango de los valores introducidos.
5. Una vez quedan las zonas en el rango, calculamos el número de contornos resultantes que van a ser tres, los leds utilizados 11.3.

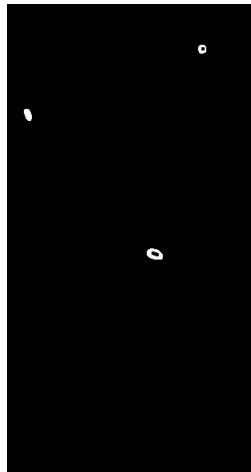


Figura 11.3: Después de aplicarlos los datos de los sliders

6. Sobre estos contornos se busca el mínimo círculo recubridor, y se dibujan, junto con líneas que van a unir sus respectivos centros en la imagen original 11.4.

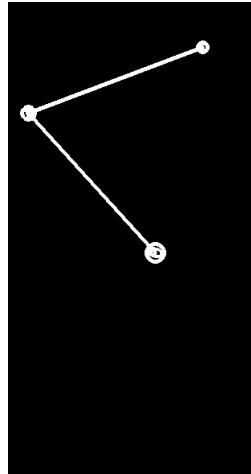


Figura 11.4: Unión de los puntos resultantes

7. Se guarda esta imagen con las modificaciones para posteriormente recrear el vídeo.
8. Sabiendo el centro de estos tres círculos se calcula el ángulo formado, y se guarda en un archivo de texto que se utilizará para generar la gráfica 11.5 y 11.6.

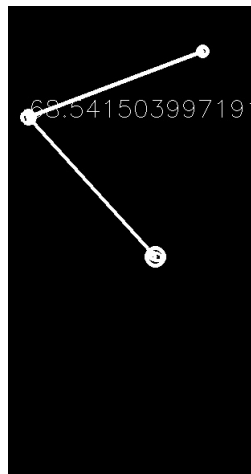


Figura 11.5: Calcular el ángulo

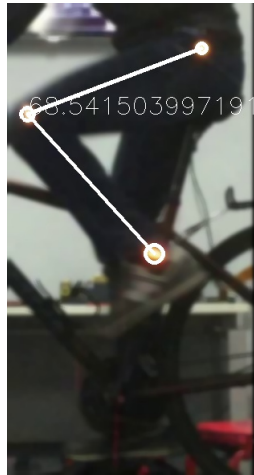


Figura 11.6: Reconstrucción de vídeo

9. Una vez analizados todos los frames del video, se leen las imágenes guardadas y se crea el mismo video pero con el dibujo de los ángulos formados. Al acabar esto se borran las imágenes para no ocupar la memoria del dispositivo.

Capítulo 12

Contribuciones al proyecto

Primeramente, es lícito resaltar que los tres integrantes estábamos interesados en contribuir y participar en todas las partes que conforman el proyecto, para así aprender cada una de ellas. Pero si es verdad, que nuestro proyecto tiene tareas a realizar muy diferenciadas, por eso para ser lo más eficientes posibles terminamos repartiéndonos las tareas en dos bloques:

1. La implementación de la cámara, permisos y realización de la interfaz gráfica de la aplicación.
2. Investigación y desarrollo en lenguaje Java del código OpenCV.

Pero como bien citamos en un principio, ninguno quería dejar de implicarse en toda la realización de AngleBike. Por esta razón, todo el desarrollo final y unión de las distintas partes lo llevamos a cabo de forma conjunta, trabajando en todo momento dentro de la facultad de informática o bien desde casa a través de aplicaciones de comunicación en tiempo real, para así aprender y conocer todos los aspectos tratados en la memoria, tanto a nivel práctico como teórico.

Así que a continuación vamos a proceder a explicaros como fue la distribución de los dos bloques a realizar:

Cuando comenzamos el trabajo, en un principio, los tres componentes del grupo nos dedicamos a investigar la librería OpenCV puesto que era lo más novedoso

para nosotros y qué nos iba a implicar más tiempo a lo largo del desempeño de nuestro trabajo.

Pero a lo largo de la realización del proyecto, mientras que íbamos realizando las reuniones con nuestros tutores de TFG, en las cuales se fijaban objetivos a realizar, nos dimos cuenta que para sacar adelante todo el trabajo y alcanzar dichos objetivos y metas que íbamos pautando, como era tanta la carga de trabajo como para realizarla conjuntamente nos dimos cuenta que nos íbamos a ver obligados por lo menos en un principio al desglose de este.

Así que procedimos a distribuir el trabajo en esos dos bloques comentados anteriormente, así que se podría decir que la parte de los mockups, la implementación de la cámara, permisos y realización de la interfaz gráfica de la aplicación han tenido más influencia, realización y peso de las decisiones de esa implementación Carlos Jaynor Márquez Torres y Laura Pérez Jambrina, quienes trabajaron en esta primera parte juntos en la Universidad.

Y para la parte de la investigación y desarrollo del lenguaje Java del código OpenCV se ha encargado más Daniel Tocino estrada, cuyo trabajo y pruebas las realizó desde casa.

Una vez dejado claro estos puntos vamos a proceder a explicar las aportaciones de una forma más detallada y clara.

12.1. Carlos Jaynor Márquez Torres y Laura Pérez Jambrina

- Importación correcta del módulo de OpenCV en Android Studio, este paso no era trivial ya que habían pocos ejemplos y los únicos que se encontraban era la importación relacionada con el plugin de Eclipse.
- Realización de los mockups, que posteriormente sirvieron de guía para el desarrollo de la implementación gráfica.
- Investigar sobre aplicaciones biomecánicas existentes en el mercado, para así conocer a qué nos exponemos y qué requisitos eran imprescindibles y mejorables de otras aplicaciones.
- Investigación y desarrollo de las pruebas sobre el funcionamiento de la cámara nativa de Android.

- Aprender a sincronizar el uso de la cámara entre la apertura de esta para una vista previa, y la apertura para una grabación con mediaRecorder.
- Implementar un código propio basándonos en las muestras que proporciona Android Developer en su página principal para poder desarrollar una cámara que permita grabar y que guarde el archivo en el formato adecuado y se adapte a la mayoría de los dispositivos del mercado. Esta ha sido quizás la tarea más difícil al ser el funcionamiento de cada dispositivo móvil distinto en sus características, como puede ser la resolución de la cámara que soporta, encontrar la extensión adecuada para que no ocupe demasiado y la vez se pueda obtener una visión adecuada para los leds, además de la versión de Android que soporta.
- La realización de los permisos de la aplicación, para que se mostrarán agrupados y no permitieran al usuario comenzar a realizar alguna de las actividades sin la activación de los que eran precisos y necesarios.
- Control del almacenamiento disponible del teléfono para que no se pueda interrumpir el comienzo de una grabación, y que además se puedan guardar correctamente el resto de ficheros que se generan con el análisis.
- La implementación de un sistema de directorios que se crean automáticamente al instalar la aplicación, los cuales cuentan con permisos de lectura y escritura, para poder almacenar en ellos y representar los vídeos, los gráficos, los frames y los vídeos resultantes del análisis en la interfaz.
- Investigación del funcionamiento de la pila de actividades y la pila de los fragmentos, ya que a medida que íbamos desarrollando la aplicación nos dimos cuenta que se formaban secuencias de ejecución que como resultado consumía memoria innecesaria de la aplicación. También fue necesario aprender el ciclo de vida de una actividad, además de los estados por los que pasa, para que la aplicación empiece y finalice correctamente.
- Investigación de Material Design para realizar una aplicación óptima y amigable para el usuario, siguiendo los patrones de profundidad, superficies, bordes, sombras y colores vistos en él.
- Realización de la interfaz de las distintas secciones que se pueden ver en la aplicación, la cuales son:
 - La parte de inicio, donde se procede a la grabación de los vídeos.

- La parte de analizar donde se muestran todas los frames resultantes de cada vídeo y que al seleccionar a uno de ellos se muestran otras dos nuevas vistas también realizadas:
 - El recorte de la sección de interés, donde se encuentran situados los leds.
 - Los filtros a aplicar a la imagen, donde se muestra la imagen ya recortada con los sliders debajo para poder ir modificando.
- La vista de la galería, donde se muestran todos los vídeos grabados y cuya selección nos lleva a otra nueva vista:
 - El reproductor de los vídeos.
- También realizamos la sección de los gráficos, donde se puede escoger entre cualquiera de los que se muestran dirigiéndonos entonces a otra nueva vista:
 - La vista de la gráfica, donde desempeñamos un gráfico en el que cada punto representa el momento en el cual se ha realizado un ángulo en concreto. Y cuya selección de uno de sus puntos nos efectuamos qué se produjera la reproducción inmediata del vídeo, en el momento y ángulo exacto escogido.
- La ejecución de la vista de los resultados, donde se muestran todos los vídeos procesados ya por el análisis y cuya selección nos lleva a otra nueva vista:
 - El reproductor de los vídeos que anteriormente ya hemos citado que habíamos creado.

12.2. Daniel Tocino Estrada

- En primer lugar me dediqué a estudiar OpenCV e investigar de qué forma se podían aislar los leds para el posterior análisis de los ángulos, por lo que empecé a hacer pruebas creando un programa Java en el ordenador.
- Una vez encontrada la forma de aislarlos y qué parámetros se necesitan cree un sistema de sliders para la configuración de color RGB que permite al usuario hacer pruebas usando el primer frame del video hasta que los leds sean lo único visible en la foto. Este código posteriormente se trasladó a la aplicación en Android.

- Al descubrir el coste del procesamiento de la imagen decidí que era mejor hacer un recorte del vídeo para seleccionar el área de intereses, es decir, la zona donde van a estar los leds durante el video con el fin de ahorrar memoria y agilizar el proceso. Para ello añadí al programa un paso previo donde con el ratón se podía seleccionar esta zona. Aunque esto no se pudo traspasar a Android por la diferencia entre pulsación en pantalla y utilización del ratón, el tratamiento de la zona seleccionada si que se mantuvo.
- El siguiente paso en el programa de ordenador fue realizar el cálculo de los ángulos del video frame a frame, que nos permitía saber qué ángulo forma el ciclista en cada momento del vídeo. Estos datos se guardan en un fichero de texto para su posterior tratamiento.
- Finalmente con el video original y los datos obtenidos se genera un video mostrando al usuario unas líneas que unen los leds y el ángulo que van formando.
- Una vez construido el esqueleto de la aplicación en Java para ordenador y teniendo mis compañeros creada ya la parte gráfica básica de la aplicación móvil comencé a integrar mi código dentro del creado por mis compañeros.
- Al volcar el código a Android me fui dando cuenta de secciones críticas que en el ordenador funcionan, pero en un dispositivo móvil no, ya mencionadas en la sección de problemas, por ejemplo la lectura del video. En este momento me puse a investigar las alternativas y soluciones posibles.
- Una vez solucionados los problemas, y con la aplicación funcionando en Android cree un adaptador para que los datos obtenidos del análisis se pudieran trasladar a la sección de las gráficas hecha por mis compañeros.
- Finalmente observando los tiempos de espera y el consumo de memoria me dediqué a la optimización de la aplicación para reducir la carga que tiene esta sobre el dispositivo móvil.

Aunque el trabajo se ha dividido en dos secciones muy diferenciadas como bien se ha redactado, con cada miembro aprendiendo de lo que estaba trabajando, en la hora de la puesta en común nos fuimos reuniendo varias veces por semana para realizar el trabajo juntos en la facultad, aun teniendo tiempo de desarrollo individual aparte.

Esta puesta en común permitió que todos los miembros acabaran comprendiendo la parte sobre la que no habían estado trabajando ya que fuimos compartiendo toda la información recolectada y resolviéndonos dudas mutuamente.

Capítulo 13

Líneas de trabajo futuro

Pensamos que cada vez son más las personas que montan en bicicleta por diversos motivos, placer, deporte, como medio de transporte o para trabajar, por ello consideramos importante mejorar cualquier aspecto de nuestra aplicación con tal de evitar lesiones futuras en la población y garantizar la satisfacción a la hora de utilizar la bicicleta, por lo que basándonos en la importancia que nosotros le damos a este tipo de deporte y a la prevención de las lesiones que este conlleva por su mala realización hay muchas líneas de trabajo futuro que puedan servir para mejorar lo ya creado o para añadir nuevas funcionalidades a la aplicación móvil.

Por tanto, en este capítulo describiremos algunas de las mejoras más relevantes que hemos encontrado entre las que se encuentran:

- Optimizar el proceso de análisis usando paralelización para reducir el tiempo de espera del usuario aprovechando todos los núcleos del dispositivo móvil.
- Cambiar el espacio de color del análisis de RGB a HSV para mejorar la precisión de detección, ya que nos dimos cuenta al final del proceso que de esta forma era más sencillo aislar los leds.
- Conseguir realizar un zoom durante la grabación del video para poder ahorrarnos el paso de recorte de la imagen.
- Realizar pruebas de la aplicación con más dispositivos android de los que hemos tenido disponibles durante el desarrollo para hacer testing.

- Hacer una evaluación con usuarios para conseguir feedback de los posibles usuarios de nuestra aplicación.
- Traducción de la aplicación al inglés para que más gente pueda acceder a ella.
- Incluir la opción de borrar los archivos creados por la aplicación desde esta misma sin necesidad de entrar al explorador de Android.
- Añadir un temporizador para la grabación del vídeo, para que así solo se precise para la realización del análisis: los leds, el rodillo y una sola persona que sería el ciclista al cual se va a analizar.
- Conseguir la detección de leds en tiempo real para facilitar el posterior análisis.

Capítulo 14

Conclusiones

Nuestro proyecto se embarcó primeramente con la investigación de los existentes estudios biomecánicos que se realizaban en la actualidad con el objetivo de conocer más de cerca cuáles eran los objetivos a alcanzar y las carencias que encontrábamos en lo ya existente. Esto nos favoreció para poder ser capaces de establecer una serie de requisitos y puntos imprescindibles a la hora de desarrollar nuestra aplicación.

Tampoco podemos olvidar, que aparte de realizar una aplicación portátil, que fuera útil y que ayudara a todos los ciclistas allí donde se encuentren, otro de los incentivos y objetivos de este proyecto era primeramente, lograr aprender a desarrollar aplicaciones Android en un entorno como es Android Studio, el cual era completamente nuevo y desconocido para nosotros y también llegar a comprender y comprender cómo funciona el tratamiento de imágenes con la librería OpenCV.

Ahora al llegar al final podemos echar la vista atrás y podemos ver que todos nuestros objetivos marcados han sido alcanzados:

- Aprendimos a desarrollar una aplicación para Android desde cero, sin tener ninguno de los componentes experiencia previa en este sistema salvo la base de Java.
- Entendimos la necesidad de la utilización del Material Design proporcionado por Android y además intentamos que nuestra aplicación siguiera los patrones de profundidad, superficies, bordes, sombras y colores vistos en él. Además que también el hecho de haber investigado aplicaciones móviles nos ha dado ideas para la realización de la interfaz de nuestra aplicación.

- Hemos estudiado OpenCV y comprendido sus distintos usos tanto para procesamiento de imágenes como para realizar análisis.
- Logramos una correcta integración tanto de OpenCV como de otras librerías auxiliares que nos han ayudado a conseguir toda la funcionalidad deseadas.
- Comprendimos la diferencia tanto entre las distintas formas de representar una imagen y sus espacios de color, como los diversos contenedores y formatos para reproducción de videos.
- Vimos de primera mano la heterogeneidad de los móviles que están en el mercado, y logramos una aplicación que funcionase de la misma manera en todos los dispositivos móviles.
- Nos dimos cuenta de la importancia de la optimización de recursos y de memoria, tanto cuando desarrollamos un programa de análisis preliminar para el ordenador, como cuando lo traspasamos al dispositivo móvil.
- Por último, hemos desarrollado una aplicación que realiza una gran serie de tareas con un grado de interconexión: grabación de un video en Android, crear una estructura de almacenamiento para los datos, procesamiento de video, análisis de este y una representación gráfica de los datos obtenidos durante este.

El único objetivo que no hemos logrado completar ha sido la evaluación con usuarios ya que la versión funcional de la aplicación no se ha acabado con suficiente tiempo como para poder hacer pruebas con personas físicas.

Capítulo 15

Conclusions

We began our project primarily by researching the existing biomechanical studies done nowadays with the objective of better knowing what the objectives to reach were and the lacking features in what was already done. This helped us to establish a series of essential requisites and points to develop our application.

We cannot forget that besides developing a portable application, useful and helpful to cyclist everywhere, we also wanted to learn how to develop Android applications on an environment like Android Studio which was in the beginning new and unknown to us and how the image processing worked with the OpenCV library.

Now we have reached the end we can look back and see that our objectives have been accomplished:

- We learnt to develop an Android application from scratch without anyone of us having previous experience besides the Java base.
- We understood the necessity of using Material Design given by Android and we tried to follow the depth, superficies, borders, shadows and colors patterns seen in it. Also researching other mobile applications gave us ideas on how to do our interface.
- We have studied OpenCV and understood its different uses for image processing and analysis.
- We reached a correct integration of OpenCV and other auxiliary libraries that have helped us reach the desired functionality.

- We learned about the differences in image representations and color spaces and the various containers and formats for video playback.
- We saw firsthand the diversity in mobile phones available in the market and we achieved an application that worked the same in all of them.
- We realized the importance of resources and memory optimization, be it when we develop it first on a computer but also when we transferred it to a mobile phone.
- At last, we have developed an application that fulfills a great deal of interwoven tasks: video recording in Android, creating a storage architecture for the data, video processing and analysis and a graphical representation of the data obtained.

The only objective we have not been able to fulfill was user testing because the functional version of the application has not been completed with enough spare time to allow us to try it with possible users.

Bibliografía

- [API] Página de android developers referido a camera api. <https://developer.android.com/guide/topics/media/camera.html#capture-video>.
- [BIK] Aplicación móvil bike fit. <http://blog.bikefit.com/bikefit-apps/>.
- [BIO] Definición de biomecánica. <http://www.maquinascyberdine.com.ar/historia-biomecanica.html>.
- [BLO] Blog sobre el tratado del error "start failed: - 19". <http://davehiren.blogspot.com.es/2016/12/android-mediarecorder-start-failed-19.html>.
- [CAMA] Camera class. <https://developer.android.com/reference/android/hardware/Camera.html>.
- [CAMb] Cámara nativa de android. <https://planetared.com/2014/06/android-l-mejora-considerablemente-la-aplicacion-camara-nativa/>.
- [CON] Condromalacia rotuliana. <http://www.planificatuspedaladas.com/condromalacia-rotuliana-en-el-ciclista/>.
- [CUS] Centro custom. <http://custom4.us/new/biomecanica-del-ciclismo/servicios-biomecanicos/estudio-biomecanico-de-ciclismo/>.
- [DdT97] San Sebastián Spain Departamento de Traumatología, Nuestra Señora de Aránzazu Hospital. *Adaptation of the lumbar spine to differents positions in bicycle racing. Spine*. 1997.
- [DRA] Android material design navigation drawer. <http://www.androidhive.info/2015/04/android-getting-started-with-material-design/>.

- [ENB] Taller de estudio biomecánico enbici. <http://www.enbici.biz/>.
- [ERRa] Error "prepare failed: -1". <http://stackoverflow.com/questions/8803100/prepare-failed-1-media-recorder-in-android>.
- [ERRb] Error "start failed: -16". <http://stackoverflow.com/questions/11434972/mediarecorder-start-failed-16?rq=1>.
- [ERRc] Error "start failed: -38". <http://stackoverflow.com/questions/12675120/android-media-recorder-start-failed-38>.
- [FIT] Fitreme. bike fitting app. https://play.google.com/store/apps/details?id=org.tostada.android.formapp&hl=es_419.
- [GRAa] Configurar variantes de compilación, gradle build. <https://developer.android.com/studio/build/build-variants.html?hl=es-419>.
- [GRAb] Graphview. <http://www.android-graphview.org/>.
- [GRI] Grid view android. <https://developer.android.com/guide/topics/ui/layout/gridview.html>.
- [HOL] De holo a material design. <https://elandroidelibre.espanol.com/2017/01/de-holo-a-material-design-cual-sera-el-siguiente-cambio-de-android.html>.
- [IMPa] Importación de opencv y javacv en android studio. <https://github.com/bytedeco/javacv/issues/353>.
- [IMPb] Importancia de los estudios biomecánicos. <https://osteopatia-jorge.blogspot.com.es/>.
- [JAVa] Utilización de java para desarrollar aplicaciones móviles. https://es.wikipedia.org/wiki/Desarrollo_de_programas_para_Android.
- [JAVb] Utilización de javacv. <https://github.com/bytedeco/javacv>.
- [JUE] Juegos olímpicos, importancia del deporte. <http://www.laizquierdadiario.com/El-origen-de-los-juegos-olimpicos-antiguos>.
- [JUS] Herramienta justinmind prototyper. <https://www.justinmind.com/>.

- [LEC] El lector de videos de opencv en android solo es capaz de leer un formato determinado. <http://answers.opencv.org/question/126732/loading-video-files-using-videocapture-in-android/>.
- [LESa] Lesiones practicando ciclismo. <http://siempreenplay.com/el-ciclismo-lesiones-mas-frecuentes/>.
- [LESb] Peter lesgaft. <http://www.sld.cu/sitios/rehabilitacion-bio/temas.php?idv=18746>.
- [MAN] Importancia de android manifest. <https://developer.android.com/guide/topics/manifest/manifest-intro.html?hl=es>.
- [MAS] La versión más utilizada de android. <http://www.redusers.com/noticias/lollipop-sigue-siendo-la-version-mas-utilizada-android/>.
- [MAT] Material design. <https://material.io/guidelines/#/>.
- [MEDa] La posición ideal. <http://www.ciclismoafondo.es/mecanica/Informes/articulo/informe-la-posicion-ideal>.
- [MEDb] Mediarecorder class. <https://developer.android.com/reference/android/media/MediaRecorder.html>.
- [MET] Metodologías ágiles. <https://sg.com.mx/buzz/la-importancia-las-metodologias-agiles>.
- [NAV] Navigation bar de android. https://developer.xamarin.com/guides/android/user_interface/navigation_bar/.
- [OCU] Ocultación de status bar en la versión kitkat. <https://stackoverflow.com/questions/29069070/completely-transparent-status-bar-and-navigation-bar-on-lollipop>.
- [OPE] Página oficial de opencv. <http://opencv.org/>.
- [PROa] Creating a new opencv project in android studio. <http://blog.codeonion.com/2015/11/25/creating-a-new-opencv-project-in-android-studio/>.
- [PROb] Problemas que suceden en los proyectos. <http://www.juntadeandalucia.es/servicios/madeja/contenido/procedimiento/20>.
- [PROc] Prototipos en justinmind. https://en.wikipedia.org/wiki/Justinmind_Prototyper.

- [Pue08] José Ramón Gómez Puerto. *La importancia de los ajustes de la bicicleta en la prevención de las lesiones en el ciclismo: aplicaciones prácticas*. “Revista Andaluza de Medicina del Deporte”, 2008.
- [RET] Sistema de análisis biomecánico. <http://transizionsport.com/blog/2015/08/17/que-es-el-sistema-retul/>.
- [ROD] Soporte para apoyar la bicicleta. <http://bicilink.com/blog-ciclismo/como-elegir-un-rodillo-para-entrenamiento/>.
- [ROI] Region of interest. <https://www.techopedia.com/definition/339/region-of-interest-roi>.
- [STA] Status bar de android. <http://www.beginandroid.com/notifications.shtml>.
- [STR] Aplicación móvil strava. <https://www.strava.com/>.
- [TAB] Tabla con las lesiones más frecuentes. <http://www.efdeportes.com/efd108/analisis-de-las-lesiones-mas-frecuentes-en-mountain-bike-mtb.htm>.
- [TRI] Tienda de estudios biomecánicos trimad. <http://trimad.es/biomecanica/biomecanica-bikefitting/>.
- [UNI] Unión soviética en los juegos olímpicos. https://es.wikipedia.org/wiki/Uni%C3%B3n_Sovi%C3%A9tica_en_los_Juegos_Ol%C3%ADmpicos.
- [VEN] Ventajas de la utilización de metodologías ágiles. <http://www.i2btech.com/blog-i2b/tech-deployment/los-beneficios-de-implementar-la-metodologia-agil/>.
- [VER] Versiones más utilizadas en android. <https://andro4all.com/2017/03/android-lollipop-version-mas-utilizada>.